A Complexity Modelling Approach to Support Early Life-cycle Phases of Assembly Automation Systems

by

Bugra Alkan

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Warwick Manufacturing Group

May 2018

WARWICK
THE UNIVERSITY OF WARWICK

Contents

List of Tables

List of Figures

Acknowledgments

I would like to express my gratitude to my supervisors Prof. Robert Harrison and Dr. Daniel Vera for their invaluable advice, guidance, and encouragement.

I would also like to extend my thanks to the Automation Systems Group (ASG) members, in particular Bilal Ahmad, Mussawar Ahmad, Mu'sab Ahmad and Malarvizhi Kaniappan Chinnathai for their ideas and feedback, which have been absolutely invaluable.

I am deeply grateful to the Ministry of National Education of Turkey for providing the government scholarship which provided me the opportunity to study in the University of Warwick.

I am also grateful to my parents and friends for their endless support not only during my education but also throughout my entire life.

Declarations

This thesis is submitted to the University of Warwick in support of my application for the degree of Doctor of Philosophy. It has been composed by the author and has not been submitted in any previous application for any degree. The work presented (including data generated and data analysis) was carried out by the author. Over the course of PhD research, parts of this thesis have been published by the author and these publication are given in the list of publication at page xiii.

Bugra Alkan

List of Publications

Peer Review Journal Papers

B. Alkan, D. Vera, B. Ahmad and R. Harrison. A method to assess assembly complexity of industrial products in early design phase. *IEEE Access*, 6: 989–999, 2017. ISSN 2169-3536. doi: 10.1109/ACCESS.2017.2777406. URL http://ieeexplore.ieee.org/abstract/document/8169677/.

B. Alkan, D. Vera, M. Ahmad, B. Ahmad and R. Harrison. Complexity in manufacturing systems and its measures: a literature review. *European Journal of Industrial Engineering* , 12: 116–150, 2018. ISSN 1751-5254. doi: https://doi.org/10.1504/EJIE.2018.089883. URL https://www.inderscienceonline.com/doi/abs/10.1504/EJIE.2018.089883.

B. Alkan, D. Vera, M.K. Chinnathai and R. Harrison. Assessing complexity of component-based control architectures used in modular automation systems. *International Journal of Computer and Electrical Engineering*, 9: 393–402, 2017. ISSN: 1793-8163. doi: 10.17706/ijcee.2017.9.1.393-402. URL http://www.ijcee.org/vol9/946-T033.pdf.

B. Alkan, D. Vera and R. Harrison. A virtual engineering based approach to verify structural complexity of manufacturing systems in early design phase. *Submitted to International Journal of Production Research*, October 2018.

Peer Review Conference Papers

B. Alkan, D. Vera, M. Ahmad, B. Ahmad and R. Harrison. A Model for Complexity Assessment in Manual Assembly Operations Through Predetermined Motion Time Systems. *6th CIRP Conference on Assembly Technologies and Systems (CATS)*, 429–434, 2016. ISSN 2212-8271. doi: https://doi.org/10.

1016/j.procir.2016.02.111. URL https://www.sciencedirect.com/science/article/pii/S2212827116003930.

B. Alkan, D. Vera, M. Ahmad, B. Ahmad and R. Harrison. A Lightweight Approach for Human Factor Assessment in Virtual Assembly Designs: An Evaluation Model for Postural Risk and Metabolic Workload. *6th CIRP Conference on Assembly Technologies and Systems (CATS)*, 26–31, 2016. ISSN 2212-8271. doi: https://doi.org/10.1016/j.procir.2016.02.111. URL https://www.sciencedirect.com/science/article/pii/S2212827116003978.

B. Alkan, D. Vera, M. Ahmad, B. Ahmad and R. Harrison. Design Evaluation of Automated Manufacturing Processes Based on Complexity of Control Logic. *26th CIRP Design Conference*, 141–146, 2016. ISSN 2212-8271. doi: https://doi.org/10.1016/j.procir.2016.05.031. URL https://www.sciencedirect.com/science/article/pii/S221282711630484X.

B. Alkan, F. Yao, B. Ahmad, M.K. Chinnathai and R. Harrison. Smart AGV management system (SAMS): An intelligent re-scheduling approach for optimising mean tardiness in manufacturing shop-floor. *Submitted to 52nd CIRP Conference on Manufacturing Systems*, June 2019.

B. Alkan, D. Vera and R. Harrison. A framework to verify structural design complexity of cyber-physical production systems. *Submitted to 52nd CIRP Conference on Manufacturing Systems*, June 2019.

Peer-review Conference Papers (Co-author)

M. Ahmad, B. Alkan, D. Vera, B. Ahmad, R. Harrison, J. Meredith and A. Binder. The Use of a Complexity Model to Facilitate in the Selection of a Fuel Cell Assembly Sequence. *6th CIRP Conference on Assembly Technologies and Systems (CATS)*, 169–174, 2016. ISSN 2212-8271. doi: https://doi.org/10.1016/j.procir.2016.02.054. URL https://www.sciencedirect.com/science/article/pii/S221282711600336X.

M. Ahmad, B. Ahmad, R. Harrison, B. Alkan, D. Vera, J. Meredith and A. Binder. A Framework for Automatically Realizing Assembly Sequence Changes in a Virtual

Manufacturing Environment. *26th CIRP Design Conference*, 129–134, 2016. ISSN 2212-8271. doi: https://doi.org/10.1016/j.procir.2016.04.178. URL https://www.sciencedirect.com/science/article/pii/S221282711630419X.

M. Ahmad, B. Ahmad, B. Alkan, R. Harrison, D. Vera, J. Meredith and A. Binder. Hydrogen Fuel Cell Pick and Place Assembly Systems: Heuristic Evaluation of Reconfigurability and Suitability. *Factories of the Future in the digital environment - Proceedings of the 49th CIRP Conference on Manufacturing Systems*, 428–433, 2016. ISSN 2212-8271. doi: https://doi.org/10.1016/j.procir.2016.11.074. URL https://www.sciencedirect.com/science/article/pii/S2212827116312288.

M.K. Chinnathai, B. Alkan, and R. Harrison. Convertibility Evaluation of Automated Assembly System Designs for High Variety Production. *Complex Systems Engineering and Development Proceedings of the 27th CIRP Design Conference Cranfield University, UK 10th 12th May 2017*, 74–79, 2016. ISSN 2212-8271. doi: https://doi.org/10.1016/j.procir.2017.01.005. URL https://www.sciencedirect.com/science/article/pii/S2212827117300069.

M.K. Chinnathai, B. Alkan, and R. Harrison. Pilot to full scale production battery module assembly case study. *27th CIRP CMS*, 2018. URL https://www.cirp-cms2018.org/.

D. Kohr, M. Ahmad, B. Alkan, M.K. Chinnathai, Lukas. Budde, D. Vera, Thomas. Friedli and R. Harrison. Proposing a holistic framework for the assessment and management of manufacturing complexity through data-centric and human-centric approaches. *International Conference on Complexity, Future Information Systems and Risk*, 2018. URL http://www.insticc.org/node/TechnicalProgram/complexis/presentationDetails/66920.

F. Assad, B. Alkan, M. Ahmad, M.K. Chinnathai, E. Rushforth and R. Harrison. A framework to predict energy-related key performance indicators of production systems at virtual validation phase. *Submitted to 52nd CIRP Conference on Manufacturing Systems*, June 2019.

M.K. Chinnathai, Z. Al-Mowafy, B. Alkan, D. Vera and R. Harrison. A Framework for Pilot Line Scale-up using Digital Manufacturing. *Submitted to 52nd CIRP Conference on Manufacturing Systems*, June 2019.

Z. Al-Mowafy, B. Alkan, M.K. Chinnathai and R. Harrison. A Data-analytics Framework to Improve the Accuracy of Virtual Engineering Tools. *Submitted to 52nd CIRP Conference on Manufacturing Systems*, June 2019.

Poster Presentation

B. Alkan. Complexity of Distributed Component-Based Automation Control Systems, WMG Doctoral Research and Innovation Conference. In *WMG Doctoral Research & Innovation Conference*, 2016. doi: 10.13140/RG.2.2.23679.30880.

Abstract

A multiplicity of factors including technological innovations, dynamic operating environments, and globalisation are all believed to contribute towards the ever-increasing complexity of manufacturing systems. Even though complexity is necessary to meet functional requirements, it is important to assess and monitor it to reduce life-cycle costs by optimising designs and minimising failure modes. The main aim of this research is to develop a scientifically valid and industrially applicable complexity modelling approach to support early life-cycle phases of industrial assembly systems against undesirable implications of static design complexity. Towards this aim, a systemic complexity modelling approach inspired by the relationships defining the π electron energy in organic molecular orbitals is introduced to the domain of industrial assembly. First, the approach is applied to industrial assembly products in order to assess and control their assembly complexity during early design stages. This is a preliminary requirement, as the product design complexity heavily influences the design of process and resources of a production system. Then, the mathematical model is revisited to assess static design complexity of assembly automation systems resulting from both logical and physical architectural designs, as well as their integration into complete systems. The novel approach is used to specify and implement a complexity assessment module integrated into a virtual system design software solution, namely the vueOne virtual manufacturing tool-sets, in order to add complexity assessment as part of the set of design support and validation tools used by manufacturing engineers. The proposed design support framework is tested on a series of assembly systems with varying degrees of static complexity. The study shows that the approach can help designers/managers to better identify root causes of static complexity, and provides a systemic approach to compare alternative system designs.

Keywords: Complexity model, assembly systems, industrial automation systems, design support and optimisation.

Abbreviations

ADT	Axiomatic Design Theory
AEM	Assembly Evaluation Method
AGV	Automated Guided Vehicle
ASG	Automation Systems Group
ASM	Assembly Structure Matrix
ASG	Automation System Workbench
BD	Bifurcation Diagram
CAD	Computer Aided Drawing
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
CC	Classification and Coding
CM	Computational Mechanics
CNC	Computer Numerical Control
D	Dynamic manufacturing systems complexity
DAC	Design for Assembly/disassembly Cost-effectiveness
DFA	Design for Assembly
DFMA	Design for Manufacture and Assembly
DFS	Design for Serviceability
DSM	Design Structure Matrix

DRM Design Research Methodology

E Enumeration

FB Function Block

FD Fluid Dynamics

FMS Flexible Manufacturing Systems

FR Functional Requirement

FSM Finite State Machine

GNT Graph and Network Theory

GUI Graphical User Interface

HP Human Perceptions

HVEMS High Volume E-Machine Machine Supply

IBD Internal Block Diagram

IEC The International Electrotechnical Commission

ISMC Internal Static Manufacturing Complexity

ISA The International Society of Automation

IT Internet Technologies

KDCM Knowledge Driven Configurable Manufacturing

KLZ Kolmogorov Lempel-Ziv Algorithm

KPI Key Performance Indicator

LET Lyapunov Exponent Testing

MBE Model Based Engineering

MDM Multiple Domain Matrix

MHS Material Handling System

MLP Make Like Production

MPS Modular Production System

NB	Non-linear behaviours
OU	Operational Uncertainties
PCB	Printed Circuit Board
PLC	Programmable Logic Controller
PLM	Product Life-cycle Management
PS	Physical situation
PSR	Phase Space Reconstruction
QFD	Quality Function Deployment
R&D	Research and Development
S	Static manufacturing systems complexity
SU	Surveys
SE	Shannon Entropy
SLOC	Source Line of Code
UML	The Unified Modeling Language
VE	Virtual Engineering
VM	Virtual Manufacturing
VRML	Virtual Reality Modelling Language
WIP	Work in Progress
XML	eXtensible Markup Language

Chapter 1

Introduction

This chapter documents the need for an analytical, integrative and systemic approach to the definition and assessment of complexity in the domain of industrial assembly. It states the research objectives and outlines the methods used for achieving the proposed research aims. In the last section of the chapter, contribution of the thesis and its structure are also presented.

1.1 Research motivation

In the last century, the global manufacturing industry has been shaped by various economic, technological and socio-political progresses, socio-environmental regulations, heterogeneity and above all, globalisation of markets and increased competitiveness [Elmaraghy et al., 2012]. Consequently, new manufacturing paradigms including increased demand for high-variety production, reduced product life-cycles, and mass customization have emerged [Efthymiou et al., 2016b]. This necessitates manufacturing enterprises to constantly improve their production systems in terms of flexibility, reliability, and responsiveness to satisfy customer demands [Vrabič and Butala, 2011]. To meet production targets of increasingly complex products with higher quality requirements and reduced time to market, the manufacturing industry makes use of highly automated production systems composed of numerous sub-systems of various nature, including: machining and processing systems, material handling devices and material storage and retrieval units [Cho et al., 2009a]. While ensuring that the system is able to satisfy the rapidly changing functional requirements, complexity increases as more components and more interfaces are introduced to the system at both hardware and software levels [Chinnathai et al., 2017]. **Figure 1.1** summarises the existence and evolution of complexity in the manufacturing industry along with its cause-effect relationships.

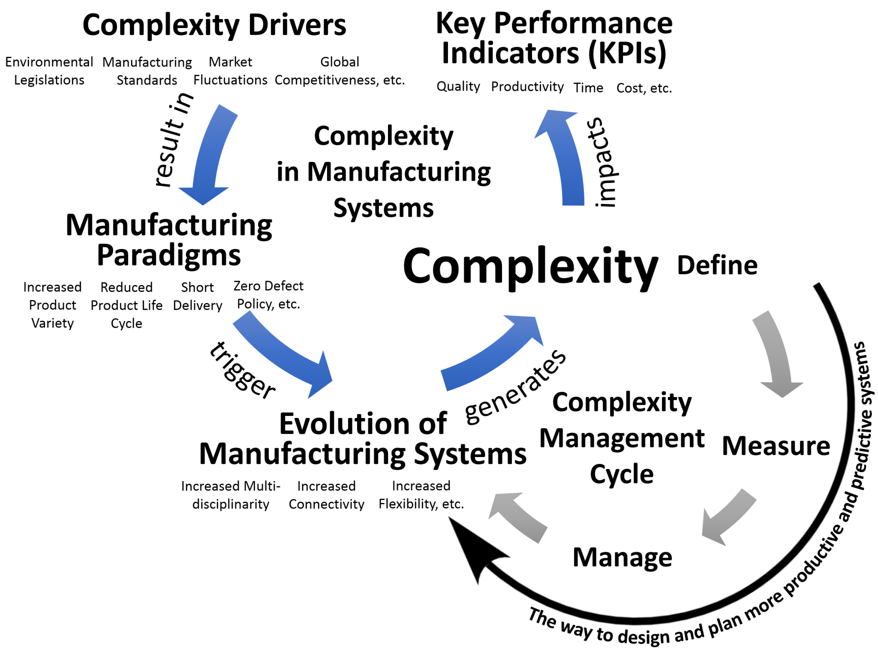

Figure 1.1: The existence and evolution of complexity in the manufacturing industry along with its cause-effect relationships (Source: [Alkan et al., 2018]).

Although complex systems may be required to enable global manufacturing requirements, complexity impacts on various factors, such as: cost, ease of reconfiguration, level of skill required across the system life-cycle (design, operate and maintain), etc. [Elmaraghy et al., 2012]. An increase in manufacturing systems complexity was reported to negatively impact all aspects of manufacturing, in terms of: production quality [Falck and Rosenqvist, 2012], reliability [Grote, 1994], throughput [Guimaraes et al., 1999; Perona and Miragliotta, 2004] and production time [Urbanic and ElMaraghy, 2006], and disturbs the systems efficiency at design, operation, maintenance, and management levels [Badrous, 2011a; Elmaraghy et al., 2012; Mattsson, 2013; Schuh et al., 2015]. Complexity leads to not only huge inefficiencies in system design and re-configuration stages but also bottlenecks in shop floor decision-making under disruptive events such as machine failures [Cho et al., 2009b]. As complexity increases, manufacturing systems become less responsive to change and harder to manage and control [Badrous, 2011b]. Moreover, complexity and the occurrence of failure within manufacturing are tightly coupled [Kinnunen, 2006; Martin, 1996; Shibata et al., 2003].

In today's manufacturing industry, complexity is considered as one of the factors inducing high cost, operational issues and increased lead time for product realisation. This conclusion is supported and verified by several industrial investigations. As an example, Collinson and Jay [2012] showed in a survey study involving more than 500 managers from over 300 companies operating in Europe, that high complexity has been designated as the cause of over 5 percent of productivity loss by votes of 63 percent of participants. In another investigation, Schleich et al. [2003] showed through a questionnaire-based study focusing on automotive companies that complexity is considered as a significant cost enabler by the majority of the applicants (**Fig 1.2**).

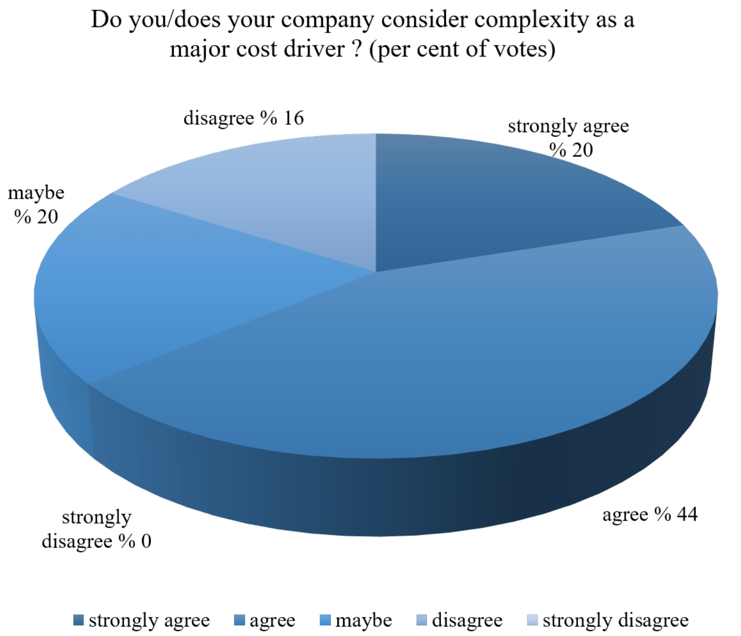

Figure 1.2: Managers consider complexity in automotive assembly systems to be a major cost driver (Source: [Schleich et al., 2003]).

Measuring complexity and trying to minimise complexity as much as possible while still meeting functional requirements or performance targets (i.e. find complexity optimum) is one way of avoiding mistakes [Alkan et al., 2016a]. According to Meyer and Lehnerd [1997], "*Reducing complexity almost always re-*

duces direct and indirect costs". Similarly, McCabe [1976b] states that measuring complexity is one of the primary requirements in a design stage of an engineering system, which helps us to better understand the cost and time required to realise it. Understanding the complexity of a design also allows us to analyse whether it is comprehensible for humans. An increase in complexity is only acceptable if it enhances capabilities, functions, usability, and performance of the system, but should otherwise be eliminated or reduced [Samy and ElMaraghy, 2012]. Therefore, complexity and its impact on the systems performance should be identified and quantified to remain profitable and competitive, and to respond rapidly to the volatile markets and rising product variety [Chryssolouris et al., 2013; Khurana, 1999; Mattsson et al., 2011a].

1.2 Research hypothesis

In an optimal design case, a manufacturing system should be designed in a way that it can satisfy all functional requirements while having the simplest possible structure, i.e. **lean system design**. In fact, the advantage of having a lean system design has been discussed by many researchers [Fisher et al., 1995; Shibata et al., 2003; Ulrich, 1995]. However, despite the recent technological advancements, the existing solutions to complexity management are still immature and typically target post-design phases of manufacturing system life-cycle, thus leading to costly and time consuming redesign phases. Thus, evaluating the root causes of complexity at the initial design state becomes an imperative implementation to design and build systems that are diagnosable, predictable and productive. These traits translate directly into reduced costs due to ease of maintenance, foresight and efficient use of resources.

It is hypothesized in this thesis work that the systematic analysis and minimisation of complexity in terms of quantifiable measures during very early design stages without compromising the required system functionality, will result in a lean manufacturing system design that provides significant benefits such as: ease of reconfiguration, ease of maintenance, increased performance, enhanced human ergonomics and improved system predictability. In order to achieve this, an analysis and assessment of complexity identifying its impacts is critical [Götzfried, 2013]. This highlights vital managerial aspects, and thus enables the development of strategies to better manage complexity, i.e. allowing designers/managers to identify the root causes of complexity and take steps to reduce and manage it. On similar grounds, a systemic approach is required to support early design phases of manufacturing systems and manufacturing processes, where complexity and corresponding critical design parameters can be identified, verified and optimised. Thus, the design and development of a systemic approach allowing rapid and accurate assessments of manufacturing systems design complexity based on industrial requirements are selected as the foci of this thesis.

1.3 Research scope

Assembly processes significantly affect products' final quality and cost [Su et al., 2010]. According to Choi et al. [2002], assembly related activities credit for more than 50% of the total production time and 20%-40% of total production cost. These findings show that assembly processes form a significant proportion of production in terms of cost and time, which implies that any improvement in assembly has direct implications on the turnover [Nof et al., 2012]. The economic importance of assembly has led to extensive efforts to improve the efficiency and cost effectiveness of assembly operations [Badrous, 2011b]. To achieve this, complexity of assembly should be identified, measured and managed [Samy and ElMaraghy, 2010b]. Measuring complexity of assembly helps designing products with ease of assembly in mind. Moreover, it helps us to rationalise various design choices of assembly processes, sequences, resources in an explicit fashion.

According to Schuh and Schwenk [2001], complexity in the context of manufacturing, can be grouped into two categories: *i*) internal and *ii*) external. Internal complexity mainly occurs as a result of high product variety due to the need to meet market demands [Chinnathai et al., 2017], whereas external complexity results from market dynamics, political and institutional complexities [Götzfried, 2013]. Internal complexity can be classified into three main category: *i*) static (structural) complexity, *ii*) dynamic (operational) complexity and *iii*) organisational (decision-making) complexity [Lindemann et al., 2008]. Static complexity is linked to the architecture of the manufacturing system, which is a network that is composed of a set of interacting resources. Dynamic complexity is driven by the manufacturing system's operational characteristics [Sinha, 2014]. Accordingly, a manufacturing system can be deemed complex, if its behaviours are difficult to describe or predict effectively [Vrabič and Butala, 2011]. It should be noted that, system behaviours are often connected to the underlying system architecture, hence dynamic complexity has a strong link with structural manufacturing systems complexity [Sinha et al., 2017]. Organisational complexity, on the other hand, is manifested in organisational structures, systems, processes and in communication flows [Kohr et al., 2017].

The scope of this research was carefully defined and focused on static (i.e. structural) complexity of assembly systems, in particular component-based assembly automation systems, which is largely under control during early design and development stages. The scope is also extended to include assembly complexity of industrial products, as it heavily influences assembly system designs. According to [Samy and ElMaraghy, 2010a], assembly system complexity is strongly linked to assembly complexity of products to be assembled. Therefore, individual components of an assembly product should be design with ease of assembly in mind, which leads to saving in both equipment and human resources. **Figure 1.3** indicates the scope of this research. Please note that, static complexity of assembly systems are analysed within the workstation level. Moreover, automotive and electronics in-

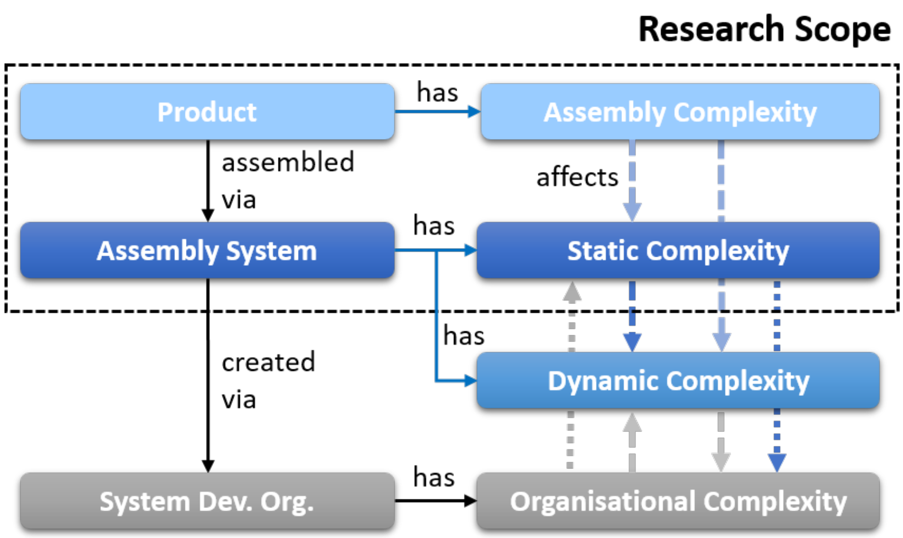

Figure 1.3: The research scope.

dustries are selected as the application domain of this research work, as this type of production relies on the design of bespoke machines and cell configuration and not of the purchase of off the shelf systems (e.g. CNC, paint shops, etc.). In addition, a number of on-going projects led by the Automation Systems Group at University of Warwick, focus specifically on the design, engineering and commissioning of automotive power-train assembly, battery modules and packs assembly, electric motor assembly in Make-Like-Production (MLP) systems which provide a plethora of engineering data that was used to feed the complexity models developed in this research.

1.4 Research aim and objectives

The main focus in this research is to make an original contribution to the management of complexity in the context of manufacturing systems engineering by fulfilling the current gap between the theoretical formulations of complexity models and their practical applicability to real-world system development. Despite the technological advancements within the last decades in manufacturing system design and development, the existing solutions to complexity management are still immature and typically target post-design phases of production system life-cycle, thus leading to costly and time consuming redesign phases. As a result, there is an increased need for tools and methods to pro-actively identify and minimise complexity during very early design stages. The research presented here, thus, aims to develop a systemic complexity model and further a proactive design support, where quantifiable data collected from virtual system design and process planning tools, can be streamlined and transformed into meaningful complexity values allowing designers to concurrently evaluate assembly system designs, to select the optimal design among various alternatives and to make modifications on existing systems. This allows designers to become aware of critical points in assembly system design which are vital in terms of system reliability and productivity. The research also aims minimise assembly system complexity by proactively identifying the assembly complexity of the industrial products that will be assembled in the system under consideration. In overall, this research presents three major objectives explained as follows:

- The first objective of this study is to understand the concept of complexity and current state of the art for developing complexity assessment approaches in the domain of manufacturing systems engineering.

- The second objective is to develop systemic mathematical models for assembly automation systems and products, where root causes of design complexity can be identified in a quantitative and repeatable fashion. This is to support manufacturers, who are engaged in tackling the problem of increased complexity, to increase system reliability and productivity.
- The third objective is to develop a design support mechanism by integrating the mathematical model into a virtual system design and development tool, where the virtual data can be automatically streamlined, and used as an input to the theoretical complexity model. This attempts to bridge the current gap between theoretical formulations of complexity and their practical applicability to real-world system development.

1.5 Dissertation outline

The dissertation is presented in a multiple manuscript format. Part of the work in Chapters 3, 5, 6, and 7 have appeared as individual research papers. The organisation of the dissertation (**Figure 1.4**) is as follows:

- **Chapter 2** reviews the concept of complexity to provide the reader with definitions, terminologies and characteristic of the concept of complexity and complex systems commonly found in the literature.
- **Chapter 3** presents a review of complexity in the domain of manufacturing engineering and its application to practical evaluation of production systems using analytic, quantitative and systematic approaches.
- **Chapter 4** introduces the theoretical basis of the research, which is borrowed from [Sinha, 2014]. The adopted analogy is further extended with a set of novel metrics, and then succesfully applied to the domain of industrial assembly in the following chapters.
- **Chapter 5** extends the approach presented in the previous chapter, and defines a complexity modelling approach which can be used in assessing product assembly complexity. The proposed approach is experimentally validated and succesfully applied to two case studies from electronics industry.

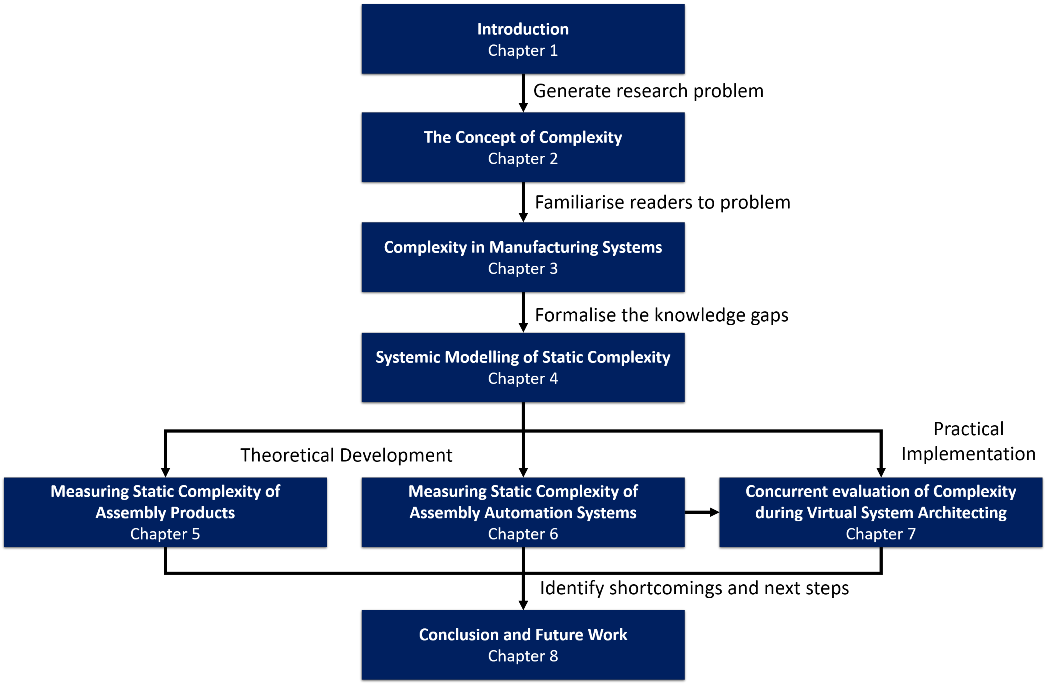

Figure 1.4: Thesis roadmap.

- **Chapter 6** extends the original method used in the Chapter 4, and proposes a theoretical framework to analyse complexity of both logical and physical architectures of modular assembly automation system designs, as well as their integration into complete systems. The proposed approach is demonstrated in a modular automation system stemming from the automotive industry.

- In **Chapter 7**, theoretical model presented in the Chapter 6 is used to specify and implement a complexity assessment module integrated into a virtual system design software solution, namely vueOne virtual manufacturing toolset, in order to add complexity assessment as part of the set of design support and validation tools used by manufacturing engineers.

- In **Chapter 8**, the results of the dissertation are discussed and the conclusions are drawn. All innovative aspects of the introduced approach are pointed out and the directions for the future work and development activities are proposed.

1.6 Chapter summary

This chapter has presented the motivation for the research work presented in this thesis. The research aim and objectives have been formally defined. The structure of the thesis has been outlined.

Chapter 2

Research background

This chapter reviews the general concept of complexity, i.e. not directly applied to the domain of manufacturing system engineering. The focus is to provide the reader with definitions, terminologies and characteristic of the concept of complexity and complex systems commonly found in the literature.

2.1 The concept of complexity

Due to the diversity of fields in which complexity is examined, a wide range of definitions are found in literature [Asan, 2009]. The word "*complexity*" is originally derived from the Latin word "*complexus*" which can mean "*entwined*" or "*twisted together*" [Elmaraghy et al., 2012]. The Cambridge English dictionary defines "*complexity*" as "*the state of having many parts and being difficult to understand or find an answer to*".

Many different approaches have been proposed to define complexity, however, a universal, precise and widely accepted terminology has not been achieved yet [Asan, 2009; Klir, 1985; Lawrence and Buss, 1994; Read, 2008; Simon, 1996; Standish, 2001]. Various discussions about complexity are focused on the basic notion of difficulty [Lee, 2003]. There is also an emphasis on the subjective nature of complexity being dependent on the system being considered [Lee, 2003] and the view of the human spectator [Gell-Mann, 1997]. As an example, Ashby [1956] introduced a relation between complexity and system scale that was defined by the relation between the system and the observer who is aiming to understand and/or control it.

2.1.1 Terminology

Several terms are used in relation to complexity and in some instances, they are used interchangeably. This section describes how these terms (simple, complicated and chaotic) are related to complexity.

- **Simple vs Complex**: The term "*simple*" is often used to express artefacts that are easily knowable and predictable, and consist of a few components [Elmaraghy et al., 2012]. For example, under the effect of gravitational force, a solid object that falls through air can be considered as a simple physical system [Suh, 2005a]. However, if the number of objects dropped in the system were to increase substantially, additional effects such as self-generated vortices may need to be considered to predict the outcome, increasing the complexity of the system [Suh, 2005a]. In other words, the knowledge and information required to define the system state has now increased in the presence of new components, and thus system become more complex.

- **Complicated vs Complex**: There is often confusion between the notions of complexity and complicatedness [Wiendahl and Scholtissek, 1994]. Systems which output unforeseeable or uncontrollable behaviour and accommodate uncertainty in their design and development processes are denoted as complex [Efthymiou et al., 2016a]. On the other hand, systems that have well-defined functions and interactions between their components and can be explained by universally well-known rules are defined as complicated [Cantamessa, 1998]. Despite these definitions of complexity and complicatedness they still do depend on the observers level of knowledge. In other words, a system which is complicated for a person, may be complex for another who has a low level of system knowledge or less advanced technological tools [Elmaraghy et al., 2012].

- **Chaotic vs Complex**: Complex and chaotic systems in the manufacturing domain are becoming a trendy research topic [Poli, 2001]. In chaotic systems, small changes in initial conditions may result in significant variations of the systems response, thus, chaotic systems may be very problematic to operate, control and maintain, and prediction of the behaviour of such systems are often impractical [Elmaraghy et al., 2012]. Reducing both the complexity and chaotic behaviours (i.e. non-linear, unpredictable) in manufacturing systems largely relies on engineering tools and methods that can make the system design and development processes more manageable and predictable (i.e. complexity management) [Elmaraghy et al., 2012].

2.1.2 Characteristics of complex systems

The conclusion from several studies [Asan, 2009; Andriani, 2003; Ashby, 1956; Casti, 1979; Cilliers, 2004; Corning, 1998; Érdi, 2008; Gell-Mann, 1997; Mitleton-Kelly, 2003; Scuricini, 1988; Waldrop, 1993; Yates, 1978] on complex systems is that they typically exhibit a number of characteristics as listed below;

- Numerousness of components can result in the system being difficult and impractical to analyse i.e. the understanding of the system functionality and its nature will be made difficult [Cilliers, 2004].
- Interconnectivity of component interactions can be of a different nature (e.g. physical interactions, information and data exchange). The complexity of the interaction themselves, coupled with the high number of interacting components are the main causes of static and dynamic system complexity and non-linear behaviours mentioned earlier in this chapter.
- Non-linearity is the main reason for small inputs causing unpredictable outputs. According to Asan [2009], non-linearity contains indeterminism (unpredictable system states), multi-stability (alternation of the states of the system between multiple exclusive states), aperiodicity (varying behaviours after some period) and irrationality (lack of normal cause effect relationship).
- Variety of components and interactions within the system.
- Significance of interactions between components may vary in complex systems.
- Hierarchy resulting from sub-systems nested over several hierarchical levels increases the difficulty of achieving description and understanding of the system static or dynamic behaviours [Asan, 2009].

- Dynamism arising from the alteration and evolution of the system over time leads to two types of complexity namely structural and behavioural complexity. Note that the structure of a system might change as part of its behaviour.

- Open Systems in which individual components can interact with the external environment defined by the system boundaries. Therefore, identifying system-context interaction (i.e. defining complex system boundaries) is often impractical and sometime impossible.

- Emergence of unexpected behaviours arising from the interactions of relatively simple components [Asan, 2009] that respond only to local information without knowledge of the system as a whole.

- Non-equilibrium causing unstable conditions that are taking the system away from its predetermined state and continuously changing it. This is a common behaviour of systems that are driven by external interventions.

- History of a system may be recorded i.e. states. The historical data may be used by the system to define its own behaviour. Therefore, without having dimension of time, analysis of such systems will be incomplete [Cilliers, 2004].

- Adaptation to external conditions through their ability to learn and adapt [Cilliers, 2004].

- Self-Organization refers to the ability to self-organize rather than being controlled by a centralized mechanism.

- Loop Activity can directly or indirectly feedback any activity to component.

These are characteristics of complex systems and not all may be applicable to modern manufacturing systems. According to Elmaraghy et al. [2012], some hardware, software and control tools can be adapted into system modules to provide the ability of adaptation (through artificial intelligence, adaptive control techniques, holonic agents and emergent principles).

2.1.3 Complexity and emergent properties

Complex systems are often defined as systems that are composed of a large number of interconnected components that as a whole exhibits one or more properties that are not evident from the properties of the individual components [Jianbo, 2013]. Such properties or behaviours are together called as *emergent property* of the system which may result in both positive and negative outcomes [Sinha, 2014]. The likelihood of emergence is established in the system architecture through the intricate causal relations across system components, which is also known as "*interconnectivity*". Hence, it can be thought as an inherent property of the underlying system architecture arising as a consequence of the combination of the system topology and behavioural aspects of individual system components [Sinha, 2014]. Emergent properties can be categorised into *i*) weak emergent properties, and *ii*) strong emergent properties [Sommerville, 2004]. Weak emergence arises when all system components work together to achieve a common objective and represents new properties arising in systems as a consequence of the interactions at an elemental level [Sinha, 2014]. Strong emergent properties, on the other hand, defines the qualities of a high-level system which are not directly traceable or irreducible to the individual system components [Laughlin, 2005]. In such a way, the whole system is different/greater than the sum of its parts.

2.2 Complexity in engineering system development

Complexity has a strong positive correlation with difficulty, as the more complex a system, the more difficult it is to design, maintain, and use, intuitively, the more difficult a task, the more expensive and error prone it is [Rechtin, 1991]. In addition to the requirements of a large amount of time for designing and integrating components, complex systems have intricate topologies or patters which may result in reduced productivity and increased failure rates during their design and development stages [Sinha and de Weck, 2013]. Similarly, Meyer and Lehnerd [1997] argues that "*Reducing complexity almost always reduces direct and indirect cost. Every additional part requires that it be made or purchased requiring time, people and capital.*"

One way of preventing mistakes is to assess and reduce complexity without compromising functional requirements and performance targets as much as possible [Sinha, 2014]. According to McCabe [1976b], assessing complexity of a design is vital in terms of predicting the cost and time essential to realise the design. Assessing of complexity also makes it visible whether the design as such is comprehensible for humans [Sinha and de Weck, 2013]. As an example, empirical studies showed that there is a strong positive correlation between complexity and number of defects found on software systems development projects (**Figure 2.1**).

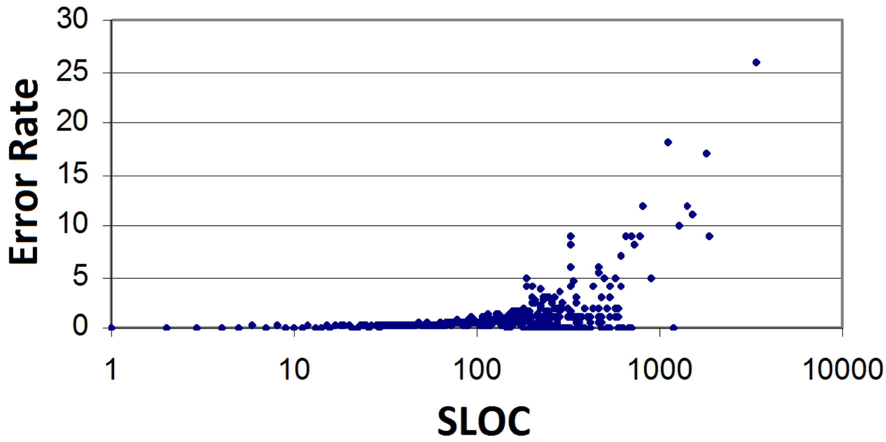

Figure 2.1: Empirical data on error-rate vs sources lines of code (SLOC) complexity metrics for software systems (Source: [Chapman and Solomon, 2002])(SLOC is a quantitative indicator of software complexity measuring the program size by counting the total lines of the program source code).

It is important to assess complexity when improving the existing products or systems. Although the goal might be to reduce existing complexity, changes in functionality of a system could adversely impact the complexity. One technique of reducing complexity is by reduction in the number of components which could in turn reduce the production cost and effort. A good example in this context, is that of the re-design of the ram air door assembly on the MD-11, the Douglas commercial aircraft company's seat airliner (please see [Ashley, 1995] for more details). The ram air door assembly is a passage for allowing air to enter the cabin's air conditioning system. It was composed of 2172 parts making it extremely complex to install. As a result, the ram sir door assembly was redesigned using principles from Design for Manufacturing and Assembly (DFMA) [Boothroyd and Alting, 1992] principles and after about two weeks the following results were noted; there was a 36 percent reduction in the number of parts and the number of assembly operations were reduced from 4038 to 2649. Moreover, the weight of the aircraft was reduced by a significant 48.53 kg. Consequently, the new ram air door design was considered more reliable and easy to maintain.

2.2.1 Complexity and multi-disciplinarity

Sometimes, the system development process can be perceived as complex, although the system under-development is well understood by the observers. This mainly occurs due to the intrinsic multidisciplinarity of the development process, e.g. number of teams, contractors, or the number of tasks in the development schedule, etc. [Sinha et al., 2017; Elmaraghy et al., 2012].

In engineering domain, multidisciplinarity is a natural result of today's agile production paradigms, since to satisfy customer demands, products and corresponding processes require integration of multiple disciplines [Motyka et al., 2006]. As an example, design and development processes of manufacturing systems are required to blend different teams from different disciplines working together and collaborating with each other. Expertise of these teams can be listed as; mechanical, electrical, electronics, control, software, industrial, artificial intelligence and further business management, human resources, quality control, stock control and many more. Although the multidisciplinarity in system development is considered as an innovation source and high added value, it, undoubtedly, increases both complexity of the development process and ultimately, chance of project failures [Tomiyama et al., 2007; Jauregui-Becker et al., 2008].

According to Tomiyama et al. [2007], complexity caused by the multidisciplinarity of system development is quite different from other complexity types, because it is an outcome of how well our knowledge is organised. Multi-disciplinary complexity problems cannot be solved by methods which are only available for a single discipline e.g. divide and conquer principle, etc. Therefore, it is required to have a set of theories, each of which is authentic for a single domain. According to Elmaraghy et al. [2012], in principle, these set of theories are independent from each other, but they are connected with inherent interactions which specify cross-disciplinary problems.

The complexity existed in systems through multidisciplinarity is clarified by investigating the set of interactions among corresponding theories, which is an approach called intrinsic complexity of multi-disciplinary developed by [Tomiyama et al., 2007]. Multidisciplinarity complexity, in fact, is caused by these interactions and the solution of this type complexity can be very problematical and time-consuming [Tomiyama, 2006]. The primary reason for this is, these interactions within an engineering system are not always observable and they may be overlooked by system designers, engineers and/or maintainers with insufficient experience. Accordingly, intensity of multidisciplinarity affects re-design and re-configuration phases of engineering systems in terms of cost and time. Therefore, effective management of multidisciplinarity complexity should be pursued and continuous awareness of the system interactions is required to be maintained.

2.2.2 Complexity and human cognition

Complexity can be understood resulting from the inherent cognitive capability of an observer, which can be termed as "*perceived complexity*" [Schlindwein and Ison, 2004]. Perceived complexity depends on the observer capability to solve, comprehend and handle the system under consideration, and hence, is different than actual complexity, which is an intrinsic property of the system. Accordingly, a system may be perceived more complex than its actual complexity by an observer who lacks of knowledge and/or technological tools [Elmaraghy et al., 2012]. There are several factors affecting perceived complexity. According to Li and Wieringa [2000], these factors include: actual system complexity, personal factors, training, experience, creativity, degree of willingness to be involved, personal type, human-system interface complexity, etc.

One way to minimise perceived complexity, in engineering domain, is the effective use of IT solutions (**Fig. 2.2**). Good examples are CAD, CAM, CAE systems which help designers to outline, operate, and compute design information in an easy and agile manner [Elmaraghy et al., 2012]. Moreover, discerning perceived complexity from the actual complexity enhances the precision by which systems can be defined, examined and certain classes of KPIs (cost, quality, performance, etc.) be foreseen [Sinha, 2014]. Therefore, analysing complexity based on quantitative metrics supports in examining the relationships between complexity and human cognitive capability in an explicit fashion.

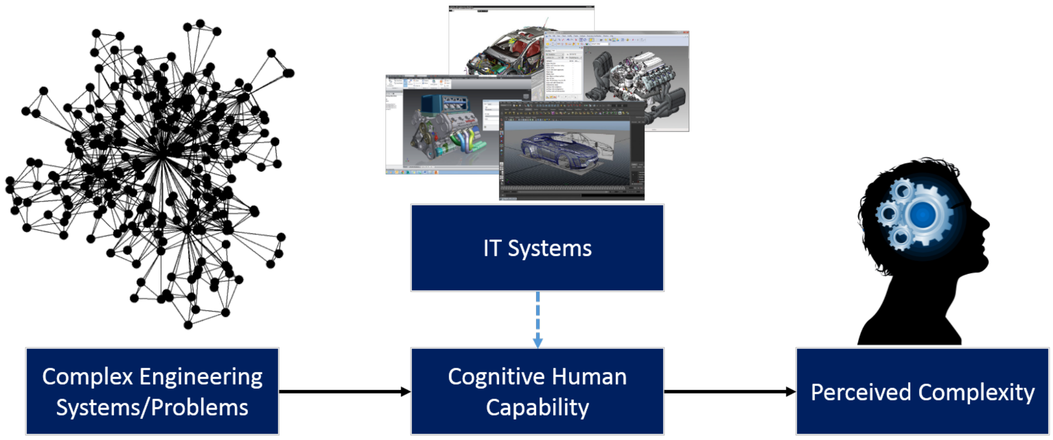

Figure 2.2: Model based engineering helps to reduce perceived complexity during complex engineering problem solving (Source: [Elmaraghy et al., 2012]).

2.2.3 Complexity and modularity

Even distribution of complexity across the system is one of the critical factors leading to the successful development of the system [Sinha and de Weck, 2013]. According to [Lankford, 2003], if a subsystem is more complex than the rest, there is a high probability that the development and maintenance processes of the system will be more costly. This is particularly correct, if the attentions is allocated to the resources regardless of the distribution of system complexity [Sinha et al., 2017]. Consequently, designers and managers should aim for a more aforethought or even distribution of complexity across the entire system [Sinha and de Weck, 2013].

There is a general belief that complexity and modularity are adversely associated, and increased modularity leads to reduced complexity. However, a highly modular system with very complex modules may also result in a complex architecture. An increase in modularity may lead to an increase of overall complexity, as the modularity may bring non-essential interfaces required to be implemented. However, this may still be wanted if system decomposability is of primary importance [Sinha, 2014]. According to Sinha and Suh [2018], "*this essentially reports to the practicality of using reductionist strategies, which humans have reasonably mastered over the last century for developing complex engineered systems*". If we take a look at Sinha [2014]'s complexity modularity trade-space framework (**Fig. 2.3**), the ideal quadrant is the low complexity high modularity zone. This is where reductionist strategies work the best and one can use decomposition to better handle

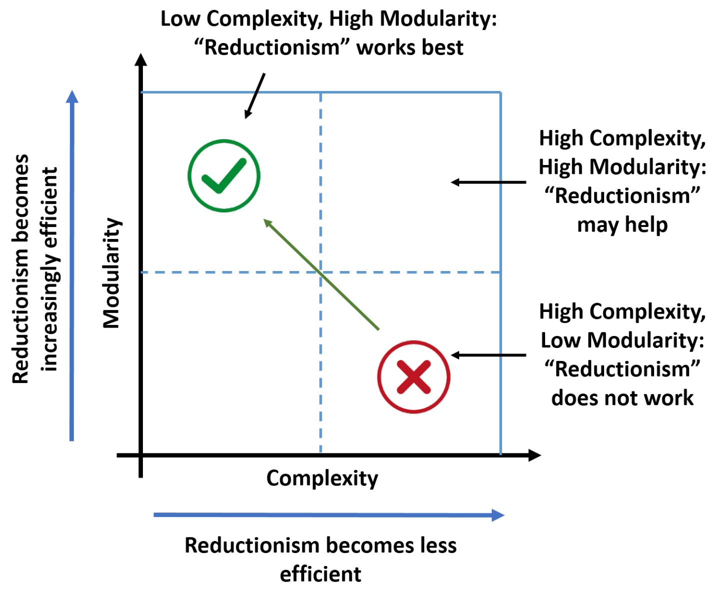

Figure 2.3: Complexity-modularity trade-space (Source: [Sinha, 2014]).

the system design and development.

In conclusion, modularisation or more specifically design encapsulation, is not inevitably a means of decreasing actual intrinsic complexity of the system, but it is a means of effectively reallocating the complexity among system components, such that the perceived complexity can be controlled and managed. This, in fact, completely lines up with our capability to *divide and conquer* [Sinha, 2014].

2.2.4 Evolution of complexity during design stages

Figure 2.4 shows the evolution of complexity during various life-cycle stages of system development. In here, essential complexity is the basic level of complexity that is required such that the system can meet its functional requirements [Sinha, 2014]. However, there is no specific method to determine the essential complexity of a given system or functional requirements. On the other hand, actual complexity is the inherent property of a system and its value, in theory, is always higher than the essential complexity, the difference of which is referred to as *excess complexity* by [Sinha, 2014]. In general, complexity cannot be managed in an effective manner, beyond a certain point determining the limit of understanding of an individual or team who is involved in the system development process. Perceived complex-

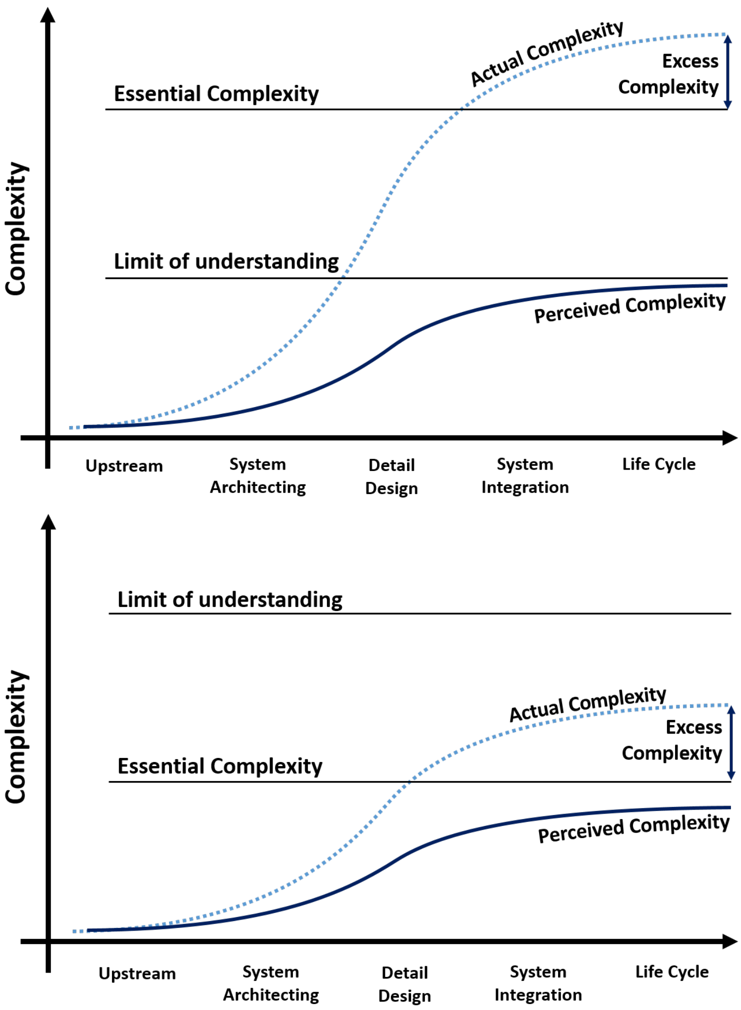

Figure 2.4: Evolution of complexity during development of a complex system (Source: [Sinha, 2014])

ity, or complicatedness, is subjective phenomenon depending on the observer of the system and is thought of to be less than both the actual and essential complexity of a system. In the initial design phases, actual complexity of the system is usually under-estimated as the actual complexity increases with the abstraction level during the system development process [Sinha, 2014]. As a result, when the system matures over time, the actual complexity of the system is revealed which could result in unprecedented situation that could lead to exceeding project budgets, missing deadlines etc. Therefore, it is highly essential to measure complexity in the early development stages in a quantitative manner, such that system development project can be managed and controlled efficiently.

2.3 Chapter summary

This chapter aims to inform the readers about the concept of complexity and explains various terminologies associated with it. The cost-complexity relationship, emergence, and impact of multi-disciplinarity on complexity, are also discussed. In the next chapter, a detailed discussion of complexity in manufacturing system, followed by a review of qualitative and quantitative methods will be provided.

Chapter 3

Literature review

Complexity continues to be a challenge in manufacturing systems resulting in ever-inflating costs, operational issues and increased lead times to product realisation. In this chapter, drivers of complexity and typical symptoms of complex manufacturing systems are identified. A comprehensive review of studies published within the last two decades to assess manufacturing system complexity are presented. The key contributions of this chapter are: *i*) a classification of complexity assessment methods based on perceived complexity symptoms; *ii*) a comprehensive review of assessment methods with cross-evaluation to identify appropriate use based on available data; and *iii*) recommendations for the wider academic and industrial community, based on research trends identified in the literature, as to how complexity assessment should be addressed in the future. It is concluded that the assessment of complexity is necessary so that it can be controlled effectively, however the industry suffers from a lack of tools to support in this endeavour. It is the role of the research community to transform complexity from a scientific exercise to something that can be practiced and administered by industry.

3.1 Drivers of complexity in manufacturing

Modern manufacturing systems work in ambiguous and rapidly changing environment guided by fluctuations in global, socio-political, and economic factors [El-Maraghy et al., 2013]. They are directly influenced by the external complexity driven by demand uncertainty and volatility, technological advancements, global competition, and supplier variability [Götzfried, 2013]. These drivers can be associated and linked with the internal complexity in a company [Marti, 2007], where they are mainly leveraged by factors such as: a high number of heterogeneous customers, large product portfolios, increased product complexity, and a high number and variety of business targets [Götzfried, 2013; Ahmad et al., 2016b]. This results in increased uncertainty in manufacturing systems leading to increased information generation and unpredicted/unknown behaviours [ElMaraghy et al., 2013].

Handling demand uncertainty requires the system to react and adapt, resulting in stochastic line balancing problems. On the other hand, an increase in demand often requires more sophisticated machine design and more machines as cycle times become the focus of the manufacturing system, thus returning to the line balancing problem. High quality standards demand additional quality check processes within the manufacturing system, again increasing the number of stations, or even the complexity of a given station such that it can assess process quality. In addition, management, analysis, and appropriate exploitation of quality data all contribute to manufacturing system complexity. Uncertainty created by the product variety, is also attributed to the complexity of tasks that operators need to carry out which, if not designed correctly, can reach the cognitive and physical limits of humans [Alkan et al., 2016b]. A combination of quantitative and qualitative parameters contributes towards operator-system complexity. Quantitative aspects include the length of a sequence, the number of tools that need to be used, ergonomics, clarity of instructions, the quality requirements, and the variety of products that the operator is required to work on [Falck et al., 2014]. Qualitative aspects refer to an operator's level of training, expertise and competence, personal factors, such as: culture, background and management strategies [Liu and Li, 2012]. The interactions of these parameters can result in unpredictable behaviour which can be difficult to control [Alkan et al., 2016c].

As manufacturing system functionality increases, so too does the manufacturing control system complexity. This is due to the integration of more modules, communication protocols, and interfaces i.e. an increase in more dependencies and couplings. This, in turn, impacts on the re-usability, modifiability, interpretability, and maintenance of the control software [Phukan et al., 2005]. Complexity also affects system ramp-up and reconfiguration efficiency [ElMaraghy et al., 2013]. Moreover, complex material flow impacts the shop floor decision making efficiency by disturbing material flow smoothness, lengthening the travel time, creating workstation starvation, and increasing the possibility of bottlenecks and downtime [ElMaraghy, 2005; Elmaraghy et al., 2012; Huang, 2003]. Multi-disciplinarity is a natural result of new manufacturing paradigms, since to satisfy customer demands, products and processes require integration of multiple disciplines [Motyka et al., 2006; Chinnathai et al., 2017]. Multi-disciplinary systems typically consist of engineering domains of varied specialisations e.g. business management, human resources, quality control, stock control, and many more. Although they are considered to be a source of innovation that adds value, they face increase in both complexity and the chance of design failures [Jauregui-Becker et al., 2008; Tomiyama et al., 2007]. A high level of concurrent engineering facilitated by multi-disciplinarity, dramatically increases both product and product development complexity which in turn impacts the manufacturing system complexity [Elmaraghy et al., 2012; Komoto and Tomiyama, 2011; Tomiyama et al., 2007].

3.2 Types of complexity

Complexity in the manufacturing systems can be defined within two domains: physical and functional [Elmaraghy et al., 2012]. Complexity in the physical domain is categorised into two types: static and dynamic [Frizelle and Woodcock, 1995] (**Fig. 3.1**). Static (or structural) complexity represents time independent characteristics of a system and focuses on types of sub-systems and strength of interconnections [Deshmukh et al., 1998]. Dynamic (or operational) complexity represents systems operational characteristics and involves aspects of time and randomness [Frizelle and Suhov, 2001]. Dynamic complexity is defined as "*the expected amount of information required to describe state of a system deviating from its performance expectations due to the unpredictability*" [Elmaraghy et al., 2012].

Complexity in the functional domain is also classified into two sub-groups: time independent and time dependent [Suh, 1998]. It is used to represent emerged uncertainty while the system is performing certain tasks under pre-defined functional requirements [Suh, 2005b]. Time-independent complexity arises from non-satisfied functional requirements during the systems life cycle e.g. the designers lack of understanding and/or knowledge about the system or component interactions [Efthymiou et al., 2012; Wiendahl and Scholtissek, 1994] and the inability to cope with a large variety of components and interactions. Time-independent com-

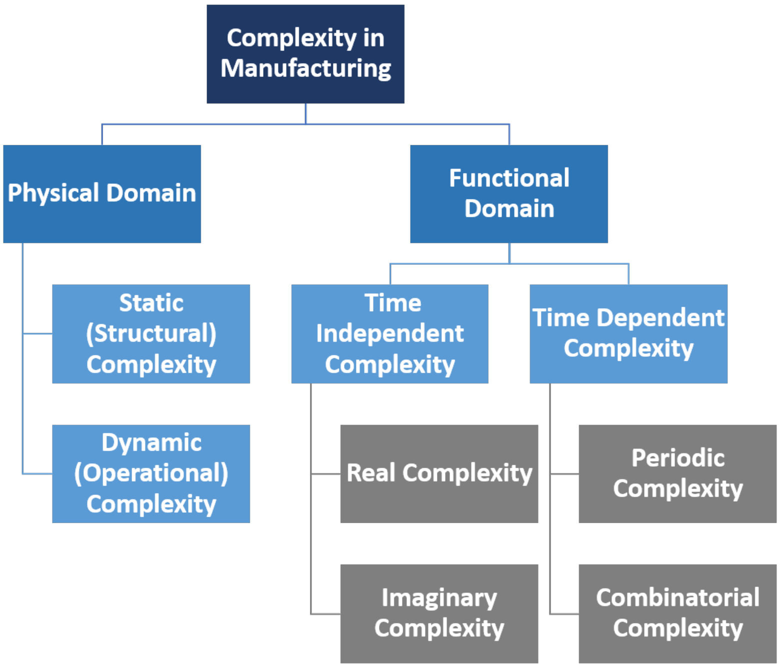

Figure 3.1: Classification of complexity types in physical and functional domains (adapted from [Elmaraghy et al., 2012]) [The figures defining physical and functional complexity types in the original document are merged in the presented figure].

plexity is further categorised into real and imaginary. Time-independent real complexity can be considered as the information content, which is a unit of probability of achieving functional requirements [Suh, 2005b]. Time-independent imaginary complexity is referred to as the unpredictability due to the lack of understanding between functional requirements and design parameters [Lee, 2003]. Time dependent complexity, on the other hand, may increase with respect to time [Chryssolouris et al., 2013]. It arises in the forms of combinatorial and periodic complexity, depending on whether unpredictability grows open-endedly or occasionally stops at a specific point and returns to the initial levels [Suh, 2005b].

There is a close relationship between systems functional complexity and its physical complexity. In engineering, it is not desired to have a simple system which is not capable of fulfilling required system functions as well as having a complex system. Without an efficient complexity evaluation, systems functionality becomes a harmful cost effector that requires to be taken into account while designing the system. According to Efatmaneshnik, M., Nilchiani, R., & Heydari [2012], logical relationship between system functionality and structural system complexity can be

Figure 3.2: Logical relationships between system functionality and system complexity.

assumed as exponential function and it is given in **Figure 3.2**. On the other hand, [Peterson et al., 2012] argues with this notion, and claims that there is no discernible trend or correlation between functionality and complexity of engineered systems. According to Peterson et al. [2012], systems with the identical levels of functionality may have large variations in its complexity. This indicates that the performance of functional execution is indicative of a given systems physical embodiment rather than the functionality itself.

3.3 Product assembly complexity

In the literature, complexity of a product assembly process is mainly studied either by analysis of product to be assembled or the process sequence for the assembly [Ahmad et al., 2016a]. The models solely based on the physical attributes of the parts are primarily influenced by approaches, by which products are designed with ease of assembly in mind, such as; Design for Assembly and Manufacture (DFMA) [Boothroyd and Alting, 1992], the Lucas Method [Chan and Salustri, 2005] and the Hitachi Assembly Evaluation Method (AEM) [Miyakawa, 1986]. Although these approaches have varied methodologies, the outcomes are similar *i.e.* reduction in part numbers, optimising part handling and insertion attributes, and penalising inefficient designs, *etc.* [Ahmad et al., 2016a]. These approaches are not intended to examine assembly complexity, instead they only attempt to enhance the product design according to the empirically verified data.

Based on an empirical study focusing on assembly deficiencies of semiconductor assembly, Hinckley [1994] found that the assembly defect rate per assembled unit is positively linked to total assembly time. His assembly complexity factor based on the Westinghouse DFA worksheet suggests a theoretical time required to assemble a product. However, this approach requires actual production data (*i.e.* the incidence of defects that occurred in the plant) and does not consider the assembly design factors which are required to evaluate the defect rates for a particular assembly station. Shibata et al. [2003] extended Hinckley's methodology and proposed an upgraded model by combining process and design based complexity factors. In Shibata's methodology, the process-based complexity factor is a function of the number of job elements in the workstation, an arbitrary threshold assembly time and time spent on individual job elements which is calculated based on the method of Sony Standard Time (SST). Design complexity factor, on the other hand, is defined as a ratio between a subjective calibration coefficient and ease of assembly results of corresponding workstation which is calculated through the Design for assembly/disassembly Cost-effectiveness (DAC). Su et al. [2010] proposed a modified Shibata's methodology which is valid for copier assembly to predict human induced assembly errors. Although these models provide a robust assessment of assembly complexity, design complexity criteria and time estimation methodologies used in these prediction models are designed for individual assembly types.

ElMaraghy and Urbanic [2004] developed an '*operational complexity index*', which is designed as a function of the quantity and diversity of both product and process elements and a relative complexity coefficient which is introduced to capture their information content. The proposed approach considers physical (*i.e.* temperature, cleanliness, envelope, strength and dexterity) and cognitive elements (*i.e.* procedures, in-process relationships and performance issues) to calculate the relative effort of each manufacturing task. Samy and ElMaraghy [2010b] extended the initial approach by adding DFA criteria to evaluate assembly complexity of individual product parts. Complexity indices are combined to acquire an overall measure for total product assembly complexity including quantity and diversity of the parts. Richardson et al. [2006] proposed a practical model to predict the difficulty of assembly of an object based solely on its physical attributes. It considers the number of components, symmetrical planes, fastenings, fastening points and novel assembly to formulate an equation which was refined using experiments in which the above-mentioned variables affect the thinking time. However, the approach is based on the data collected for a specific type of assembly, therefore, requires further work to produce the definitive model.

3.4 Symptoms of complex manufacturing systems

Analysing and understanding manufacturing complexity allows us to develop and implement the correct strategies for management of complexity [Efthymiou et al., 2016b]. Based on the observations from the literature, different definitions for the conceptualisation of specific aspects of complexity in manufacturing systems have been found. These definitions distinguish manufacturing system complexity based on a number of symptoms to indicate its existence. Accordingly, twelve symptoms that are perceived to be an indication of complexity have been identified (according to existing literature within this domain). These symptoms are then grouped into four classes which have been selected based on the perspective of observation of a given symptom: non-linear behaviours, operational uncertainties, physical situation and human perceptions, and summarised in **Table 3.1**. **Table 3.2** classifies the reviewed studies based on the complexity type, class of symptoms and the theoretical origins of the assessment method used.

3.4.1 Symptoms observed from non-linear behaviours

The most typical feature of complex systems is the existence of non-linear and chaotic behaviours [Cilliers and Spurrett, 1999]. According to Scholz-Reiter et al. [2002], complex and dynamic behaviours can be observed even in relatively simple

Table 3.1: The symptoms of complex manufacturing systems.

Class of symptoms	Symptoms used in the assessment of manufacturing system complexity
Non-linear behaviours	The existence of repeating patterns observed in the long term behaviours
	Sensitivity to the initial demand and production control parameters
	High impact of structural modifications on the manufacturing performance
Operational uncertainties	Increased information content of resource states and process scheduling queues
	Significant deviations between scheduled and observed resource states
	Uncertainty in handling product variety
	Stochasticity and unpredictability of manufacturing processes and system KPIs
	Existence of the turbulence in the manufacturing flow
Physical situation	Increased diversity, quantity and information content of system related elements
	High dependency and interconnectivity between system related elements
Human perceptions	Knowledge complexity
	Technological complexity

manufacturing systems. In the literature, a number of studies identify complexity in the existence of symptoms associated with irregular dynamic phenomena whose identification require scanning of production records over a reasonable time interval.

The first symptom in this class, is the existence of irregularities and dominant patterns observed in the long term behaviours of production systems. In this context, long term behaviours indicating the interaction and evolution of dynamic system parameters which are defined by geometrical structures generated through phase space reconstruction methods, such as: time delays and recurrence plots. This symptom is a result of dynamic complexity and investigated in the following studies [Chryssolouris et al., 2004; Deif and ElMaraghy, 2009; Donner et al., 2008; Giannelos et al., 2007; Wiendahl and Scheffczyk, 1999; Katzorke and Pikovsky, 2000].

The second symptom is the sensitivity to initial conditions, and emergence of chaos. Accordingly, systems exhibiting large deviations in meeting due dates or performance goals by even small changes in initial conditions or production control parameters, such as WIP levels, can be considered as complex. This symptom is a result of both static and dynamic complexity resulting from the factors such as: production delays, multiple-feedback loops and external and internal disturbances, and analysed through chaos and non-linear dynamics methods, such as bifurcation diagrams and maximal Lyapunov exponent testing in the following studies: [Alfaro and Sepulveda, 2006; Donner et al., 2008; Massotte, 1996; Papakostas and Mourtzis, 2007; Schmitz et al., 2002; Scholz-Reiter et al., 2002].

Table 3.2: Review of the literature on manufacturing system complexity.

Authors (year)	Type		Class of symptom				Theoretical origin of the method used										
	S	D	NB	PS	OU	HP	SE	KLZ	CM	FD	LET	PSR	BD	E	CC	GNT	SU
Frizelle and Woodcock [1995]	●	●			●		●										
Massotte [1996]		●	●										●				
Sarkis [1997]	●			●										●			
Deshmukh et al. [1998]	●				●		●										
Calinescu et al. [1998]	●	●			●	●	●										●
Wiendahl and Scheffczyk [1999]		●	●									●					
Kim [1999]	●	●		●		●								●		●	●
Guimaraes et al. [1999]	●	●				●											●
Katzorke and Pikovsky [2000]		●	●									●					
Frizelle and Suhov [2001]	●	●			●		●										
Huaccho Huatuco et al. [2001]		●			●		●										
Schmitz et al. [2002]	●	●	●									●					
Scholz-Reiter et al. [2002]		●	●										●				
Calinescu [2002b]	●	●			●	●	●										●
Efstathiou et al. [2002]	●	●			●	●	●										●
Sivadasan et al. [2002]		●			●		●										
Fujimoto et al. [2003a]	●				●		●										
Makui and Aryanezhad [2003]	●				●		●										
ElMaraghy and Urbanic [2003]	●	●		●											●		
Peters et al. [2004]		●	●										●				
Chryssolouris et al. [2004]		●	●									●					
ElMaraghy and Urbanic [2004]		●		●											●		
Wang et al. [2005]		●	●								●						
ElMaraghy et al. [2005]	●			●	●		●								●		
Phukan et al. [2005]	●			●										●			
Zhang and Efstathiou [2006]	●	●			●		●										
Alfaro and Sepulveda [2006]		●	●								●						
Kuzgunkaya and ElMaraghy [2006]	●			●	●		●								●		
Urbanic and ElMaraghy [2006]	●	●		●											●		
ElMaraghy [2006]	●			●											●		
Rao and Efstathiou [2006]	●	●			●	●	●										
Wu et al. [2007b]		●			●		●										
Gabriel [2007b]	●			●										●			
Zhu et al. [2007a]	●				●		●										
Papakostas and Mourtzis [2007]		●	●								●		●				
Giannelos et al. [2007]		●	●									●					
Liu et al. [2008]	●				●		●										
Frizelle and Suhov [2008]		●			●			●									
Zhu et al. [2008]	●				●		●										
Hu et al. [2008]	●				●		●										
Windt et al. [2008]	●	●		●										●			
Donner et al. [2008]		●	●									●					
Papakostas et al. [2009]	●	●	●								●						
Deif and ElMaraghy [2009]		●	●									●					
Schleifenbaum et al. [2009b]	●	●			●					●							
Romano [2009]	●	●			●					●							
Huaccho Huatuco et al. [2009]		●			●		●										

Table 3.2: Review of the literature on manufacturing system complexity (continue).

Authors (year)	Type		Class of symptom				Theoretical origin of the method used										
	S	D	NB	PS	OU	HP	SE	KLZ	CM	FD	LET	PSR	BD	E	CC	GNT	SU
Cho et al. [2009a]	●				●		●										
Efthymiou et al. [2009]	●	●			●					●							
Maksimović and Petrović [2009]	●			●												●	
Zhu [2009]	●				●		●										
Wang et al. [2009]	●				●		●										
Jenab and Liu [2010]	●			●											●	●	
Wang and Hu [2010]	●				●		●										
Wang [2010]	●				●		●										
Sivadasan et al. [2010]		●			●		●										
Abad [2010]	●				●		●										
Garbie and Shikdar [2010]	●	●		●										●			
Han et al. [2011]	●			●										●			
Abad and Jin [2011]	●				●		●										
Wang et al. [2011]	●				●		●										
Vrabič and Butala [2011]		●			●				●								
Zhang [2011]	●	●			●		●										
Mattsson et al. [2011b]	●	●				●											●
Espinoza et al. [2012]	●			●												●	
Vrabič and Butala [2012]		●			●				●								
Zhang [2012]	●	●			●		●										
Samy and ElMaraghy [2012]	●			●											●		
Fässberg et al. [2012]	●				●		●										
Ding and Sun [2012]	●	●			●		●										
Mattsson et al. [2012]	●	●				●											
Gabriel [2013]	●			●										●			
Wang et al. [2013]	●				●		●										
Chryssolouris et al. [2013]	●	●		●	●		●	●									
Mourtzis et al. [2013]		●			●			●									
Fast-Berglund et al. [2013]	●				●		●										
Smart et al. [2013]	●	●			●		●										
Mattsson [2013]	●	●				●											●
Zeltzer et al. [2013]	●	●		●										●			
Efthymiou et al. [2014]		●			●			●									
Elmaraghy et al. [2014]	●			●												●	
Mattsson et al. [2014]	●	●				●											●
Schoettl et al. [2014]	●			●										●			
Park and Okudan Kremer [2015]	●				●		●										
Liu et al. [2015b]	●			●											●		
Samy et al. [2015]	●			●											●	●	
Alkan et al. [2016a]	●			●	●		●									●	
Thomé and Sousa [2016]	●	●				●											●
Lukáš and Plevný [2016]		●			●		●										
Modrak and Bednar [2016]	●				●		●										
Mattsson et al. [2016b]	●	●				●											●
Zeltzer et al. [2017]		●			●		●										
Presented study	●			●												●	

The third symptom in this class is the dynamic behaviours emerging from the coupling between the intrinsic structure of the system and uncertainty related to its operations. This symptom is a reflection of static complexity occurring due to the structural alterations (e.g. adding/removing equipment) and analysed via bifurcation diagrams and maximal Lyapunov exponent testing in [Papakostas et al., 2009] and [Schmitz et al., 2002], respectively.

3.4.2 Symptoms observed via operational uncertainties

An increase in complexity results in various operational problems including batch-and-queue decision-making inefficiency, lack of process synchronisation, increased lead and ramp-up times, and performance fluctuations. In the literature, a number of symptoms observed through uncertainties in the operational flow, is used to perceive manufacturing system complexity.

The first symptom in this class is the increased amount of information required to describe the scheduled state of the system and its components. This symptom is a reflection of intrinsic difficulty of the operation for producing the required number and type of products in a certain period of time [Calinescu et al., 2000], and it arises due to the various factors, including: increased number of parts, operations and machines, increased sequence flexibility, and increased resource sharing, etc.[Deshmukh, 1993]. In this context, information content is linked to the uncertainty associated with the probability of an entity being in a predefined state. For example, in case of a machine, states can be defined as busy, idle or in maintenance and their probability can be measured through production order and process plans for individual parts [Calinescu, 2002a]. This symptom is a direct result of static complexity and is analysed by means of Shannon entropy in the following studies: [Calinescu et al., 2000; Deshmukh et al., 1998; Efstathiou et al., 2002; Frizelle and Suhov, 2008; Frizelle and Woodcock, 1995; Huaccho Huatuco et al., 2001; Liu et al., 2008; Makui and Aryanezhad, 2003; Park and Okudan Kremer, 2015; Zhang and Efstathiou, 2006; Zhang, 2011, 2012].

The second symptom is the operational dynamism occurring due to several factors such as: part reject, rework, absenteeism, and resource breakdowns, etc. [Calinescu, 2002a]. Accordingly, systems in which it is difficult to monitor their operational status, can be considered as complex [Frizelle and Woodcock, 1995]. In this context, complexity is estimated by analysing the deviation between observed and scheduled resource states (i.e. the probability of a resource being out of control) which is captured through real-time process observations taken at regular intervals. This symptom is a consequence of dynamic complexity and investigated by means of Shannon entropy in numerous studies [Alfaro and Sepulveda, 2006; Calinescu et al., 1998; Calinescu, 2002a; Frizelle and Suhov, 2001; Frizelle and Woodcock, 1995; Huaccho Huatuco et al., 2009; Sivadasan et al., 2010; Smart et al., 2013; Wu et al., 2007b; Zhang and Efstathiou, 2006; Zhang, 2011, 2012].

The third symptom in this class is the uncertainty in handling increased product variety which is often linked to the risk factors associated with operator's choices of tools, fixtures, and assembly procedures. In this context, complexity is referred as the averaged uncertainty in a random process of handling product variety, which depends on the sum of the feed varieties at the station and the transferred varieties from all the upstream stations. This symptom, also referred to as the operator choice complexity, is a representation of static complexity associated with the system configuration topology, and investigated by means of Shannon entropy in the following studies: [Fast-Berglund et al., 2013; Wang and Hu, 2010; Wang, 2010; Wang et al., 2011, 2013; Zhu et al., 2007b, 2008; Zhu, 2009].

The fourth symptom in this class is the degree of uncertainty associated to the predictability of manufacturing operations and system KPIs. This symptom is a consequence of dynamic complexity occurring due to the factors such as: incompleteness of information, disturbances, and uncertainties inherent to the manufacturing environment, and analysed through two different approaches.

The first approach captures this symptom by analysing the prediction efficiency of manufacturing processes. In this approach, complexity is linked to the averaged historical memory stored in the process, and computed by the computational mechanics approach which employs Shannon entropy over the distribution of causal states of a production machine recorded over a reasonable time interval. [Vrabič and Butala, 2011, 2012] (see [Shalizi and Crutchfield, 2001] for the concept of causal states). The latter analyses unpredictability of manufacturing system KPIs by employing the Lempel-Ziv analysis of finite sequences [Chryssolouris et al., 2013; Efthymiou et al., 2014; Mourtzis et al., 2013].

The last symptom in this class is the existence of manufacturing flow turbulence arising due to the relations between lead time, system performance, process structure and production system configuration. In the literature, this symptom is analysed by employing the concept of Reynold number derived from fluid dynamics analogy in the following studies [Efthymiou et al., 2009; Schleifenbaum et al., 2009a; Romano, 2009].

3.4.3 Symptoms observed from the physical situation

Complex system theory defines a complex system as a system which is composed of many components and exhibits hierarchy and self-organization arising due to the dynamic interaction of its components [Bar-Yam, 1997]. From this viewpoint, the third class contains complexity symptoms that can be perceived through analysing system's physical situation: (*i*) increased variety, quantity and information content of system elements, and (*ii*) the significance of their interrelations and interdependencies. These symptoms can be searched within the various aspects of the system, e.g. system configuration, material flow patterns, control and information flow patterns, intrinsic process hierarchy, etc., and analysed by means of heuristics-indices based approaches including: enumeration and classification and coding, as well as the methods derived from graph theory.

In this context, enumeration based approaches try to capture information content of the system by counting system-related elements, e.g. resources, products, customer orders, tasks, etc., in a systematic manner [Garbie and Shikdar, 2010; Kim, 1999; Sarkis, 1997; Schoettl et al., 2014; Windt et al., 2008]. In this group, a number of studies also attempted to link complexity to the system performance by correlating the enumerated elements with the real or simulated production data [Gabriel, 2007a, 2013; Han et al., 2011; Zeltzer et al., 2013].

Classification and coding based approaches, on the other hand, finds the relative importance of each enumerated factors by means of heuristics based classifications. As an example, a pioneer approach (i.e. structural classification coding [ElMaraghy, 2005]) developed by the research group at the University of Windsor, classifies the various types of system equipment (e.g. machines, material handling systems, buffers) based on the amount and variety of information required to use, operate, programme, control and interact them [ElMaraghy and Urbanic, 2003; ElMaraghy, 2005; Liu et al., 2015a; Samy et al., 2015; Samy and ElMaraghy, 2012; Urbanic and ElMaraghy, 2006]. These approaches are also used together with the

Shannon entropy to link complexity to the uncertainty associated with the information content of resource states (e.g. resource availability) in [ElMaraghy et al., 2005; Kuzgunkaya and ElMaraghy, 2006].

In the literature, a number of studies, also, perceives complexity as the system's information content characterised by the connectivity and dependency among system elements (e.g. material flow connections and dependency within the process hierarchy etc.). These approaches often use methods derived from graph theory (e.g. node betweenness centrality, vertex degree, etc.) [Alkan et al., 2016a; Elmaraghy et al., 2014; Espinoza et al., 2012; Maksimović and Petrović, 2009; Samy et al., 2015; Chryssolouris et al., 2013; Modrak et al., 2013; Modrak and Bednar, 2016].

3.4.4 Symptoms observed from human perceptions

Along with its objectivity, complexity also has a subjective nature, making it dependent on the system being considered [Lee, 2003] and the view of the human spectator [Gell-Mann, 1997]. In view of that, the last class of symptoms contains complexity indicators which can be perceived by humans. In this class, the symptoms are classified into two sub-groups: (*i*) technological complexity indicating the complexity of the underlying technology used to perform system related activities and (*ii*) knowledge complexity representing the domain-specific knowledge and decision-making complexity. In the related literature, individual perspectives about manufacturing system complexity are analysed and captured using subjective assessment methods such as: structured and semi-structured questionnaires, surveys and interviews [Calinescu et al., 1998; Calinescu, 2002a; Guimaraes et al., 1999; Kim, 1999; Mattsson et al., 2011a; Mattsson, 2013; Mattsson et al., 2012, 2014, 2016a; Thomé and Sousa, 2016].

3.5 Methods for assessing manufacturing complexity

By following a classification scheme mainly based on the taxonomy presented by [Efthymiou et al., 2016a], these methods are investigated according to respective theoretical origins: (*i*) chaos and non-linear dynamics theory, (*ii*) information theory, (*iii*) heuristics, (*iv*) graph theory (*v*) fluid dynamics analogy, (*vi*) surveys, and (*vii*) hybrid methods. **Figure 3.3** shows the complexity symptom-assessment method pairings.

3.5.1 Chaos and non-linear dynamics theory

Chaos and non-linear dynamics system theory is a trending mathematical area with increasing interests in the fields of physics, engineering and social sciences. Chaos theory provides a robust theoretical framework for understanding non-linearity, uncertainty and instability, and it is considered to be a well-established science. In the literature, the methods derived from chaos and non-linear dynamics theory are often employed to measure complexity through analysing symptoms connected to the system's dynamic behaviours. These methods include: phase space reconstruction, maximal Lyapunov exponent testing, and bifurcation diagrams.

3.5.1.1 Phase space reconstruction

Phase space reconstruction aims to construct the system state through using corresponding historical data and observing it in a higher dimensional space [Rong-Yi and Xiao-Jing, 2011]. Reconstruction of a phase space can be done via several different ways, such as: phase portraits, Poincare map, recurrence plots and time delay plots. Phase portraits are two dimensional projection of trajectories of a dynamic

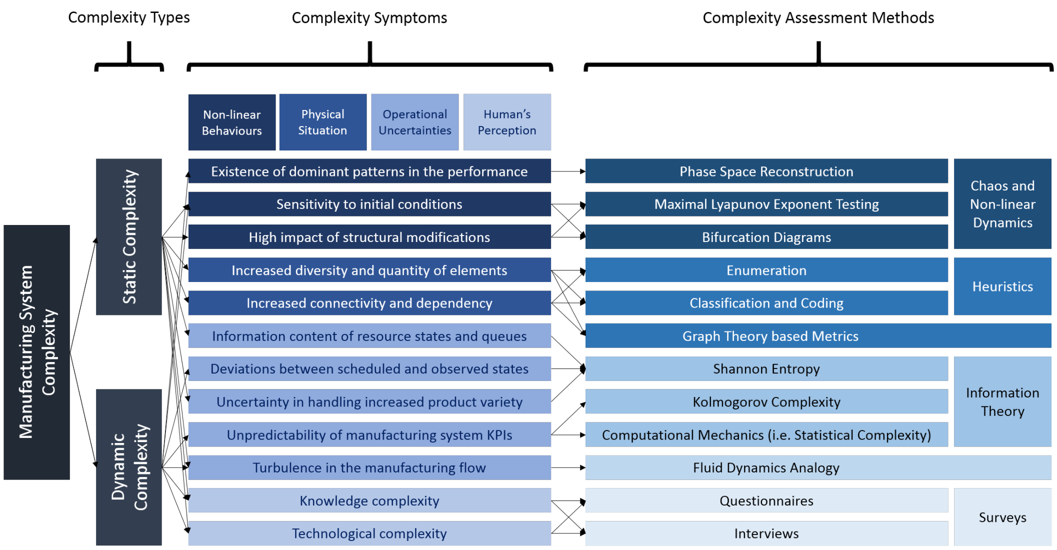

Figure 3.3: The complexity symptom-assessment method pairings.

system in the phase space which are useful predominantly for system visualization. A Poincare map is a one dimensional map generated by extracting data points from the intersection of a phase space trajectory with a lower dimensional space [Herbst and Herzel, 2013]. Recurrence plots are a two-dimensional representation scheme aiming to detect hidden dynamical patterns and non-linearities through bringing out distance correlations in time series.

Phase space reconstruction techniques provide a deeper understanding of system behaviours and corresponding factors that contribute towards behavioural changes by offering views of the system in geometric patterns. They have been employed in several studies aiming to analyse non-linear and unpredictable behaviours of modern production systems. Wiendahl and Scheffczyk [1999] investigated a simple simulated model of a paint-spraying system. The paint thickness of a coating depending on the previous layer of coating and machine parameters such as adjustable spray gun pressure, are investigated. The employed control function was found to cause deterministic chaotic behaviours inducing a unique pattern in the phase space while being undetectable in statistical analyses. Katzorke and Pikovsky [2000] inspected a simple balanced three-funnel model (explained in [Wiendahl, 1987]) of production dynamics for both continuous and discrete order flows. Peters et al. [2004] examined dynamical behaviours of an idealized manufacturing system subjected to the interaction of scheduling policies and buffer capacity restrictions. Poincare map and bifurcation diagrams were employed in these analyses. According to the authors, there is a direct relationship between dynamical behaviour of a manufacturing system and its production performance. Chryssolouris et al. [2004] and Giannelos et al. [2007] studied dynamic behaviours of dispatching rules in a simple manufacturing system using phase portraits and time delay plots, respectively. Donner et al. [2008] studied dynamics of a logistic network consisting of a low number of cooperating manufacturers through discrete event simulations and the recurrence plots phase space reconstruction. Deif and ElMaraghy [2009] adapted system dynamics approach to study the dynamic capacity scalability in multi-stage manufacturing systems associated with the operational complexity of the capacity scaling processes. Complexity was defined as the required effort calculated in terms of magnitude and frequency of the capacity scaling response in dynamic demand.

3.5.1.2 Maximal Lyapunov exponent testing

Maximal Lyapunov exponents testing studies the exponential rate of divergence or convergence of trajectories starting from nearby initial points in phase space and hence, is primarily used to study the sensitivity and dependency of dynamic systems on their initial conditions [Sandri, 1996]. Systems that have at least one positive maximal Lyapunov exponent are considered sensitive and chaotic. This method offers valuable insights on how the stability of manufacturing systems can be influenced by structural and operational variations and therefore provide a broad understanding of the chaotic nature of systems.

In the literature, several studies have analysed chaotic and non-linear behaviours of manufacturing systems using maximal Lyapunov exponents testing. Massotte [1996] examined chaotic behaviours of a simple closed loop system. Wang et al. [2005] proposed a methodology to analyse dynamic behaviours of a system to achieve better lot-sizing decisions. Schmitz et al. [2002] surveyed chaotic behaviours on discrete manufacturing systems. Alfaro and Sepulveda [2006] proposed a step by step methodology to estimate system sensitiveness to initial conditions. Papakostas and Mourtzis [2007] analysed the adaptability of a manufacturing system subjected to demand varieties.

3.5.1.3 Bifurcation diagrams

Bifurcation diagrams allow the comprehension of how the long term behaviours of a system change, as particular variables fluctuate. Discontinuities and bifurcations in the diagram point out the changes in the system behaviours [Efthymiou, 2013]. Due to their ability to capture unstable and unexpected behavioural changes and to identify critical system parameters that lead to an unwanted change in the behaviours, these diagrams are considered an effective methodology and have been implemented in a number of studies that investigate the sensitivity of manufacturing systems' performance on design changes [Papakostas et al., 2009]. Scholz-Reiter et al. [2002] studied irregular behaviours of a production system which was assumed to be a part of more complex facility. A set of possible control methods were suggested which are valid for different levels of WIP by using non-linear dynamics methods. Papakostas and Mourtzis [2007] investigated the dependence of production rate of a steel production company on specific values of model parameters.

3.5.1.4 Limitations of methods derived from chaos and non-linear dynamics theory

According to Efthymiou et al. [2016a], this approach offers valuable understandings of the system behaviours, visualises the effect of system parameters on the key performance indicators, and depicts the sensitivity of the system. However, a set of limitations has been flagged in the related literature. Modern manufacturing systems often exhibit stochastic events (e.g. machine breakdowns) rather than deterministic chaos. However, tools and methods developed based on this theory, are not able to capture and analyse such stochastic events [Efthymiou, 2013]. Moreover, only maximal Lyapunov exponents testing provides a quantitative measure for chaos within the manufacturing system, other methodologies are limited and offer only schematic analysis for the dynamic system behaviours [Efthymiou et al., 2012]. Furthermore, the approaches used for approximation of the Lyapunov exponents require relatively big data sets and they are highly sensitive to the fluctuations in the external factors such as measurement errors and noise [Efthymiou, 2013].

In summary, theory of chaos and non-linear dynamics can be considered a highly valuable tool in behavioural analysis of manufacturing systems, revealing dominant patterns in manufacturing system performance. However, these methods require a costly measurement phase and they are not able to capture stochastic complexity sources, therefore it is still questionable as to whether these tools are a practical solution for real industrial environments.

3.5.2 Information theory

Information theory, principally proposed in Shannon's study of communication theory (please see its revised version in [Shannon, 2001]), considers entropy as the degree of ambiguity associated to the outcomes of a random experiment. In other words, entropy is described as the degree of disorder within a system [Fast, 1962]. In the manufacturing domain, this approach is used to capture the following symptoms (*i*) scheduling and observation based information content of resource or queue states, (*ii*) deviation between scheduled and actual states of the resources, (*iii*) uncertainty in handling product variety with the context of risk factors related to the operator choices, and (*iv*) unpredictability of manufacturing processes and manufacturing performance indicators.

3.5.2.1 Shannon entropy

In manufacturing domain, Shannon entropy is used to quantify the uncertainty of identifying the required information to define the state of a manufacturing system or its components. As an example, it defines static complexity of a manufacturing system H_S as the amount of information required to define the state of the production system and it can be formulated as follows;

$$H_S = -\sum_{j=1}^{M}\sum_{i=1}^{N} p_{ij} \log_2 p_{ij} \tag{3.1}$$

where, M is the quantity of resource existing in system S, N is the number of possible states for the j^{th} resource and p_{ij} is the probability of state i occurring in resource j. In this context, states of resources can be defined subjectively (e.g. busy, idle and breakdown etc.). The probability of states can be measured based on scheduling information (static complexity) and real time observations (dynamic complexity). Frizelle and Woodcock [1995] used Shannon entropy to optimise operation strategies of a manufacturing enterprise. Static complexity was measured through focusing chiefly on queue lengths, whereas dynamic complexity was calculated based on observed states of manufacturing resources (i.e. idle, busy or failed). Deshmukh et al. [1998] enumerated the factors affecting the static complexity of a manufacturing system to define a static complexity metric related to processing requirements and machine capabilities. In the paper, the variation in static complexity was investigated in terms of part resemblance, system volume and product design alterations (**Fig. 3.4**). Frizelle and Suhov [2001] proposed an entropic complexity measurement to assess the rate of variety in queueing systems and networks by employing Kolmogorov-Sinai entropy (see [Petersen, 1983]). Calinescu et al. [1998] proposed a comparison between entropy and questionnaire based complexity assessment measures. The proposed entropy measure accounted for the following factors: product structure, the structure of shop or plant, planning and scheduling functions, information flow and dynamism, and variability and uncertainty of environment. Efstathiou et al. [2002] presented an expert system to evaluate the decision-making complexity of system-organization interactions that used existing

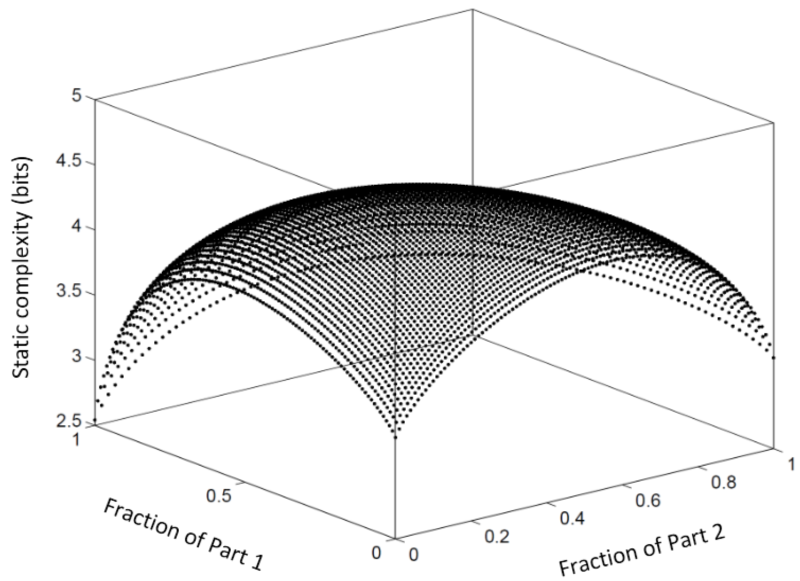

Figure 3.4: The relationship between static complexity and different part mix ratios (Source: [Deshmukh et al., 1998]).

company data to compute complexity and offer recommendations. Frizelle and Suhov [2008] developed a method to assess dynamic complexity by evaluating the evolution of manufacturing queueing lengths and resource state conditions in three different case studies.

Shannon entropy is also used to measure complexity related to the deviations between scheduled and observed resource states. Huaccho Huatuco et al. [2001] investigated scheduling complexity in a bottle supplier enterprise. Dynamic complexity was assessed by estimating conditional probabilities associated with deviating scheduling states and it was found that complexity can be varied with both customer demand changes and organizational flexibility. Sivadasan et al. [2002] proposed a metric for supplier-customer networks based on the uncertainty of material and information. This metric is then extended and validated in [Sivadasan et al., 2006]. Wu et al. [2007a] surveyed the relationship between operational complexity and inventory costs. Huaccho Huatuco et al. [2009] proposed a comparison between five different rescheduling strategies based on their effectiveness in reducing complexity that arises due to stochastic machine breakdowns. A series of simulations were performed for this purpose which accounted for: overall information content, variations between schedules, and mean flow time. Reducing unbalanced machine workloads and using low disruption strategies were suggested to reduce operational complexity. Sivadasan et al. [2010] examined the relationship between networks of customers and suppliers, and operational complexity (e.g. scheduling variations). An increase in the operational complexity was found to have a significant association with the reduction in the supplier's inventory capacity. It was suggested that operational complexity could be better managed by incorporating: IT systems, shorter scheduling planning and more frequent information exchanges.

Another implementation of Shannon entropy solely focuses on the assessment of the uncertainty in handling increased product variety associated with the risk factors related to the operator choices. Fujimoto et al. [2003b] developed an information theoretic method to interpret product variety induced complexity arising at the different stages of assembly system by utilising weighted Shannon entropy. In each assembly station, information entropy was assumed to originate from two kind of aspects: (*i*) variety flowing through a station (*ii*) product varieties adding in the station. Zhang and Efstathiou [2006] proposed a complexity metric based on the Shannon entropy for mass customisation in manufacturing systems. The authors pointed out that complexity arises from inventory management primarily influenced by the number of stock locations and the number of product variants that are stored in these areas. Zhu et al. [2007b] introduced a measure called "*operator choice complexity*" in order to pursue optimal assembly sequences in mixed-model assembly lines by reducing process sequence complexity which in turn reduced system complexity.

Zhu et al. [2008] surveyed operator choice complexity by consolidating product mix and process information in mixed model assembly systems, and provided guidelines for managing complexity at the design phase of such systems. Cho et al. [2009a] developed a quantitative complexity assessment approach for various configurations of assembly and disassembly stations. The proposed approach uses probability distribution of information associated with the part processing times, part mix ratios and routings. Wang and Hu [2010] investigated the relationship between system throughput and complexity associated with human related factors such as operator reaction time and fatigue effects. According to the findings, product variety induced complexity affects the reliability rate of manufacturing stations and disturbs a station's throughput.

Hu et al. [2008] proposed a measure for both assembly systems and their supply chains. The proposed measure has three classifications. First, station level complexity was related to the summation of the entropies calculated from the product mix ratios associated to the sequential assembly activities at corresponding stations. Second, system level complexity was defined as the summation of the complexity occured due to product varieties introduced at the stations and from the upstream stations. Third, assembly supply chain complexity was determined by considering three different factors, namely; (*i*) supply chain configuration, (*ii*) product variety in each node and (*iii*) demand uncertainty in each node. The study was revisited and extended by Zhu et al. [2007b] to examine the complexity of supply chain configurations and their relationship with assembly systems. Wang et al. [2011] carried out an optimisation study focusing on the relation between product variation and complexity in semi-automatic assembly systems. In this study, a novel measure called "*relative complexity*" based on a theoretical model [Makui and Aryanezhad, 2003] was developed to find out the optimal set of variants to enlarge the market share while reducing complexity. Wang et al. [2013] extended a previous complexity model [Wang et al., 2011] and carried out an optimisation study focusing on the configurations of the mixed model assembly systems.

3.5.2.2 Kolmogorov complexity and Lempel-Ziv analysis of finite time series

Kolmogorov complexity is an application of the algorithmic information theory in computer science, named after Andrey Kolmogorov who first presented this subject in 1963. According to Kolmogorov's idea, the complexity of any binary string is the size of the smallest binary computer program that can reproduce this string on the Universal Turing Machine and then halt [Cover and Thomas, 2006]. Lempel-Ziv complexity metric [Lempel and Ziv, 1976], on the other hand is a non-parametric scale of finite sequences and it has been used in several applications, including coding and lossless data compression. This metric is presented based on Kolmogorov's axioms and it is associated to the quantity of diverse substrings and the proportion of their existence along a given sequence [Efthymiou et al., 2014].

In recent years, this metric has been extensively applied in manufacturing systems and manufacturing supply networks to evaluate the irregularity of manufacturing KPIs. The research group at the University of Patras has published a number of complexity assessment studies based on Kolmogorov complexity and Lempel-Ziv analysis of finite sequences [Chryssolouris et al., 2013; Efthymiou et al., 2014; Mourtzis et al., 2013; Papakostas and Chryssolouris, 2011]. One of the pioneer works presented by [Efthymiou et al., 2014], investigated the unpredictability of performance indicators in manufacturing systems. In the study, the fluctuations in the performance time series of critical manufacturing indicators generated through discrete event simulations, were studied by employing Lempel-Ziv complexity measure, then an overall complexity indicator was calculated through assessing weighted average Lempel-Ziv complexity.

3.5.2.3 Computational mechanics

Computational mechanics concerns the issues of pattern, structure, and organization and producing a model of a hidden process generated from observed behaviours [Shalizi and Crutchfield, 2001]. This approach offers an information-theoretic methodology to find optimal causal models of stochastic processes. Vrabič and Butala [2011] chiefly adopted this approach in manufacturing systems by employing an information theoretic measure to assess prediction efficiency of manufacturing processes. In the work, complexity was represented by "*statistical complexity*", defined as the quantity of historical memory collected during the past processes. Historic data was used to generate symbolic sequences and further turned into ε-machines to capture patterns and regularities reflecting the casual structure of the process (**Fig. 3.5**). In the analysis step, statistical complexity and efficiency of the predictions were evaluated based on Shannon entropy. This approach differs from other entropy based measures as it relates unpredictability with complexity.

3.5.2.4 Limitations of information-theoretic measures

Information theoretic measures propose an objective way for quantifying both dynamic and static complexity of manufacturing systems. This approach provides a single complexity score which enables comparison of design alternatives. Nevertheless, a set of problems bound back the applicability of the information theory. According to Efthymiou et al. [2016a], information theoretic measures are insufficient to link complexity with the manufacturing system performance. Also, these measures rely on costly observation and measurement data gathering phases to calculate probability estimation of scoped aspects. In cases such as new designs without having a prototype or real system, the data required to capture such information

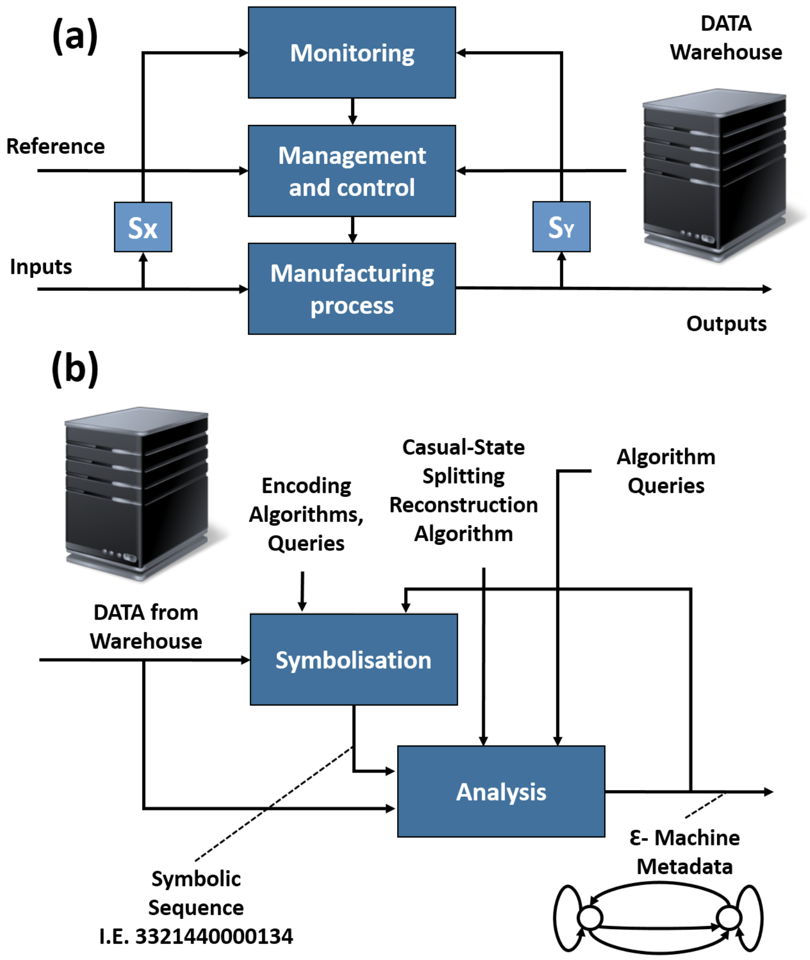

Figure 3.5: Computational mechanics approach: a) Data acquisition, warehousing, b) symbolisation and analysis (Source: [Vrabič and Butala, 2011]).

is nearly impossible. Kim [1999] stated that computer simulations might be an alternative way to calculate system capability, but in most cases it is impractical in terms of cost and time. Moreover, information theoretic complexity measures provide a single complexity value which provides an insufficient level of granularity to determine where efforts should be focused to make improvements. On the other hand, information theory includes two essential assumptions which may be critical in terms of accuracy. Kim [1999] explains the first assumption as; "*complexity is a universal quality that exists, to some degree in all objects, and there is a uniform metric for measuring the complexity of a system*". Klir [1985] argued this statement and stated that describing the complexity as an inherent attribute of an object is not purposeful from an operational point of view. The other assumption states that variables of a system are considered independent. According to the researchers [Badrous, 2011a; ElMaraghy, 2005; Elmaraghy et al., 2012], this assumption is not true for real systems which limits the applicability and accuracy of the approach. Similarly, Kim [1999] and Elmaraghy et al. [2012] pointed out that information theory complicates measurement for large systems and assumption of validity of independent states should be replaced with the conditional entropy approach. Peliti and Vulpiani [1988] pointed out that complexity should be considered as subjective and relative. Furthermore, as there is a subjectivity associated with the selection of resource and queue states, information theoretic measures may struggle to explain perceived complexity e.g. interactions between human and machine. Issues to be addressed in information theory, include the impact of defective information, measurement cost for dynamic complexity assessment, conversion of the results into meaningful information, and recommendations for issues on manufacturing system design and management [Alkan et al., 2016a]. According to Smart et al. [2013], accuracy in probability estimation, interdependency assumption, sample quality and long data recording are the most important factors to be taken care of for those who start out to gather data for measuring structural and operational complexity based on entropic approach. Moreover, further investigation is still required to enhance the predictive capabilities of the information theoretic measures. As an example, Kolmogorov complexity Lempel-Ziv analysis method heavily depended on the observed performance time series length [Efthymiou et al., 2014]. Also, this approach require a common time series length for the comparison of dynamic complexity of different manufacturing systems, which may not be the case in many situations Efthymiou et al. [2016a]. Computational mechanics approach, on the other hand, provides a promising solution to achieve a relationship between complexity and system's operational performance. However, this measure suffers in terms of practicality, as it requires relatively big amount of data necessary to analyse dynamic complexity.

3.5.3 Heuristics

As opposed to previous methods, which guarantee to give a quantifiable reflection of the system complexity, heuristic based approaches attempt to provide an industrially readable picture of complexity based on the system's physical situation. Due to their intuitive nature, heuristic methods are advantageous in that they are easily applied to real industrial systems and data collection is easy, thus allowing comparisons of design alternatives at early in the life cycle phases to detect potential stress points.

3.5.3.1 Enumeration

Approaches using enumeration relate complexity to the number of system, product and process related elements, such as: quantity and diversity of resources and manufacturing tasks as well as the number of demand changes. Sarkis [1997] studied the relationship between complexity and productivity of a flexible manufacturing system (FMS). In the study, complexity is considered as the summation of the total number of installed industrial robots and numerically controlled machines. According to the results, a continuous drop in productivity performance is observed as the system's complexity increases. This is attributed to an increase in the number of devices, which correspondingly increases the required efforts (i.e. scheduling and transportation) to operate these devices, which in turn dramatically impacts the efficiency of the system.

Kim [1999] studied the effects of product variety over system complexity by proposing a set of metrics consisting of three dynamic and static complexity sources: *i*) relationships between system components described as the quantity of flow paths, number of crossings in the flow paths, cumulative part travel distance, number of combinations of product and machine match, *ii*) inherent properties of system components, such as: number of elementary system components and complexity of each elementary component and *iii*) people related issues, such as: process improvements, information accessibility, number of suggestions, etc.

Gabriel [2007a] proposed a static complexity measure called "*Internal Static Manufacturing Complexity*" (ISMC). The ISMC is designed as a function of distinct number of manufactured components, number of work centres, the volume of production, and the commonality between different product routes. Another enumeration methodology, "*the complexity cube*", is a vector based complexity assessment approach developed by [Windt et al., 2008]. In this approach, production system complexity is considered to be a combination of time-related, organisational, and systemic complexity aspects which are represented by vector formats (**Fig. 3.6**).

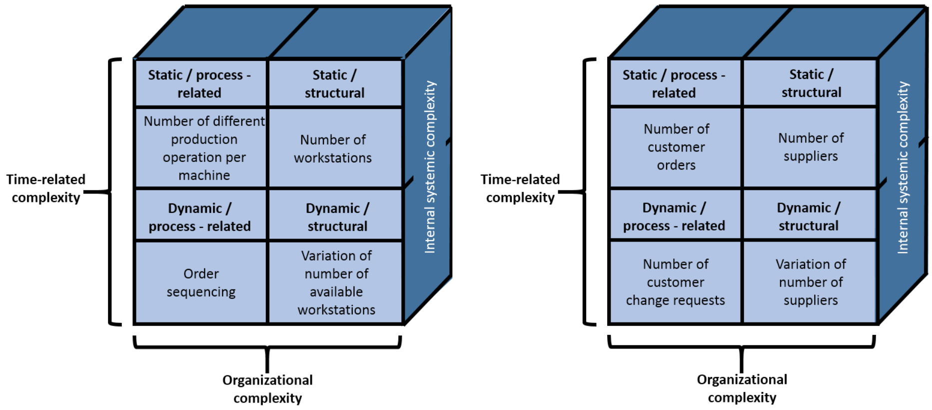

Figure 3.6: Some characteristics of the three dimensions of the complexity cube (Source: [Windt et al., 2008])

3.5.3.2 Coding and classification

The research group at the University of Windsor, Canada has proposed a coding and classification system which aims at quantifying time-independent complexity based on assessing the key aspects of manufacturing systems. ElMaraghy and Urbanic [2003] considered complexity as a combination of three key factors: *i*) the absolute amount of information, *ii*) the variety of information and *iii*) the information content linked to the exertion required to manufacture a machining feature of a product. Product complexity is represented by the product complexity index, which is calculated by counting different design parameters such as; quantity of features, quantity of inspection checks and diversity of part elements, etc. Process complexity is defined as a function of the product design, the volume requirements, planning horizon and the work environment. The proposed methodology can, according to its authors, be used in any design situation through selection of the suitable facets of the main product and process elements.

The original approach was extended by Urbanic and ElMaraghy [2006], to cover complexity in manual manufacturing operations by taking some facets of cognitive complexity related to operator perception into account. A new definition called the "*operational complexity index*" was introduced. The operational complexity index is calculated through analysing and assessing a series of indices consisting of process and product related operational information. The proposed measurable operational complexity index was considered a valuable tool when modelling human operator performance. These complexity metrics offer a hybrid measure of complexity for manufacturing operations, where several complexity indices connect information content and diversity with information entropy.

ElMaraghy et al. [2005] developed an indices based method for manufacturing systems that utilizes heuristics and information theory in which availability of each component is taken into consideration. The metric consists of different complexity fields representing inherent structural and operational characteristics of classes of entities, such as: machines, buffers, material handling systems (MHS) and operators. Each field was made up of a string of digits, where digit values represent the degree of structure, control, and operation complexity of the corresponding feature (**Fig. 3.7**). Higher digit values reflect higher complexity, in other words, a higher amount and variety of information required to operate, control, programme or interact with corresponding system module. Later, Kuzgunkaya and ElMaraghy [2006] adapted a hybrid approach to evaluate configuration complexity of reconfigurable manufacturing systems and developed a new measure. The proposed metric is calculated based on information theory, where the state probabilities are defined based on the reliability of different system modules, such as: machines, buffers and MHS. ElMaraghy et al. [2010] and further Samy and ElMaraghy [2012], extended the original classification and coding approach to include assembly oriented static complexity sources of various manufacturing system resources, including machines, buffers and MHSs.

3.5.3.3 Limitations of heuristics based approaches

Heuristics based complexity assessment approaches are close to industrial practice where they attempt to capture the overall information content of a production system using user-subjective or counting based information collection techniques. Due to this intuitive starting point, these methods provide a set of advantages. Heuristic approaches are often employed in real industrial systems due to their ease of use, simple data collection and non-expert result interpretability features. These tools can be a valuable solution when data availability is limited, i.e. very early design stages. However, as set of limitations has been issued for the heuristics approaches. Due to their subjective nature, these approaches provide a weaker vision of manufacturing system complexity and they are unable to analyse complicated connections within a system [Elmaraghy et al., 2012]. These metrics are heavily dependent

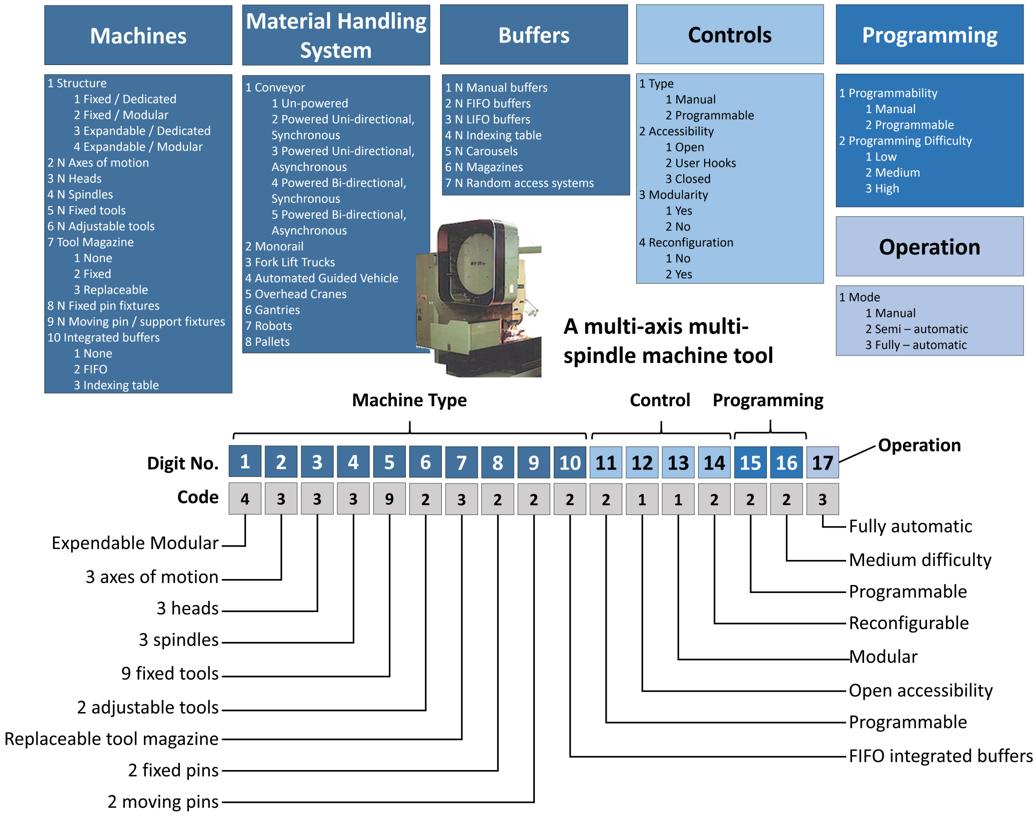

Figure 3.7: Coding and classification approach (Source: [Elmaraghy et al., 2014])

on the industrial domain or specific focus that they are designed for, thus, the applicability of heuristics based approaches over different types of production systems and focuses is often limited. In conclusion, heuristics based approaches provide an intuitive view regarding complexity associated with the physical situation, however, due to its subjective nature, it is debatable as to whether these measures reflect overall system complexity accurately.

3.5.4 Graph and network theories

Graph and network theories provide a basis for investigating the entities and their relationships within a system [Kreimeyer, 2009]. In recent years, a number of works [Alkan et al., 2016a; Chryssolouris et al., 2013; Jenab and Liu, 2010; Elmaraghy et al., 2014] that have direct and indirect utilisation of the graph and network theory in the assessment of physical aspects of manufacturing system complexity, have been proposed. Chryssolouris et al. [2013] proposed a complexity measure called "*network complexity*", in which graph theory is used to produce adjacency matrix which represents the connection between product, process, and resource domains. The vertex degree is then used to assess the coupling between these domains. ElMaraghy et al. [2014] developed a complexity model based on the graph theory which incorporates information content of the system represented by characteristics of its layout. The model was tested on different types of manufacturing systems and several guidelines for designers to reduce manufacturing system layout complexity were provided.

3.5.5 Fluid dynamics analogy

The fluid dynamics analogy in the manufacturing domain is an analytical framework which has been used previously in modelling of system performance indicators and management of scheduling issues [Asl et al., 2000; Avram et al., 1995; Dai, 1995; Weiss, 1999]. In manufacturing systems, the fluid dynamics analogy is chiefly used to analyse manufacturing flow turbulence. Efthymiou et al. [2009] used this analogy as a theoretical background and introduced the Reynold number concept to manufacturing systems which aims to identify the transition regime between steady and turbulent manufacturing operations in different work-flow conditions. Moreover, similar Reynold number concepts have been used in assessing complexity in manufacturing systems [Schleifenbaum et al., 2009a] and supply chains [Romano, 2009]. Although, fluid dynamics analogy can be considered as a promising approach for detection of critical areas that contribute to turbulence in production, it is still a premature practice and requires further investigations [Efthymiou et al., 2016a].

3.5.6 Surveys

Perceptions about complexity in a manufacturing company is often gathered using questionnaires and interviews. In the literature, questionnaires are often used to analyse the degree of usability of a specific technical tool [Williamson, 2000].

In the literature, there are a number of papers focusing on both manufacturing system complexity and complexity arising due to user-system interactions by employing structured questionnaires and interviews. Calinescu et al. [1998] proposed a metric based on Meyer and Foley Curley's management of software development framework described in [Meyer and Curlnotey, 1995]. The complexity was investigated in two different domains: knowledge complexity and technology complexity. Knowledge complexity is described by a metric consisting of seven different complexity variables. Technology complexity is considered to be the complexity of the underlying computer technology, and it is defined with a metric made up of eight different complexity variables. This method is then employed in a case study in order to assess decision-making complexity. Data used in this study is gathered by questionnaires at different levels of hierarchy within the selected company.

Mattsson et al. [2011a] developed a questionnaire based complexity index, namely: CXI, where users assess production complexity, subjectively. Questionnaire parameters are categorised into five main groups: product/variants, process methods, station layout, equipment, and organisation and environment complexity sources. Falck et al. [2012] proposed an assessment model for assembly task complexity based on the interview study suggesting criteria to identify both low and high assembly complexity. The grade of fulfilment of the aforementioned criteria is used to reflect the degree of production complexity.

A similar study which is proposed by Mattsson [2013], aims to define manufacturing complexity based on a series of structured interviews in which, subjective opinions of human workers regarding product variants, work content, layout, tools and work instructions are collected. Questionnaires, surveys and interviews attempt to provide insights on how humans perceive manufacturing systems during their life-cycle. They can be used to analyse bottlenecks and to get indications of potential improvements by flagging the interrelating complexity concerns.

Although, survey based approaches can capture the perceived level of complexity, these approaches cannot be used in the evaluation of system designs since no physical mock-up or process trials are available. Also, they are limited to questionnaire-stage and their results are dependent on the subjective interpretation of the interviewees.

3.6 Research trends

The literature review shows an increase in the total number of articles published per year (**Fig. 3.8a**); it indicates a growing trend for management and optimisation of complexity in manufacturing systems. This can again be inferred from **Fig. 3.8b**, that in the last decade, the number of published studies that discuss about complexity evaluation have increased by more than half when compared to the previous decade. It can be seen from **Fig. 3.8c**, that the importance given by academic community to static and dynamic complexity is almost equal, however, the slight increase in the focus on static complexity could be attributed to the fact that it is relatively easier to identify and assess static complexity.

It is discernible from **Fig. 3.8d**, that almost half of the publications investigate complexity symptoms associated with operational uncertainties. This signifies the increased attention given to operational efficiency in scheduling and planning, and shop floor decision-making in comparison to the other classes of symptoms. It is also important to note, studies perceiving complexity through system's physical situation also gained a significant increase. This indicates the importance of proactive evaluation of system designs at the conceptual and preliminary design phases. Also, the popularity of employed methodologies is given in **Fig. 3.8e**.

3.7 Research gaps

The primary reason for evaluating the complexity of manufacturing is to design and build systems that are diagnosable, predictable and productive. These traits translate directly into reduced costs due to ease of maintenance, foresight and efficient use of resources.

This chapter has presented studies that have been published over the last two decades that offer methodologies for measuring complexity. However, there does remain a gap between the definitions of complexity as understood by academics versus those who practice engineering in industrial environments. Approaches that examine complexity during the operational phase of manufacturing systems are often costly; they require large data sets collected by on-site observations and analysed using expert systems. These offer an ability to identify non-desirable behaviour and optimise accordingly. On the other hand, approaches (i.e. heuristics based approaches) that measure complexity from the physical situation of manufacturing systems are less successful as the large amounts of data required are not available at this point of a system's life-cycle As a result, an assessment of complexity cannot, and thus is not, typically made in industry at the design phase as managers and

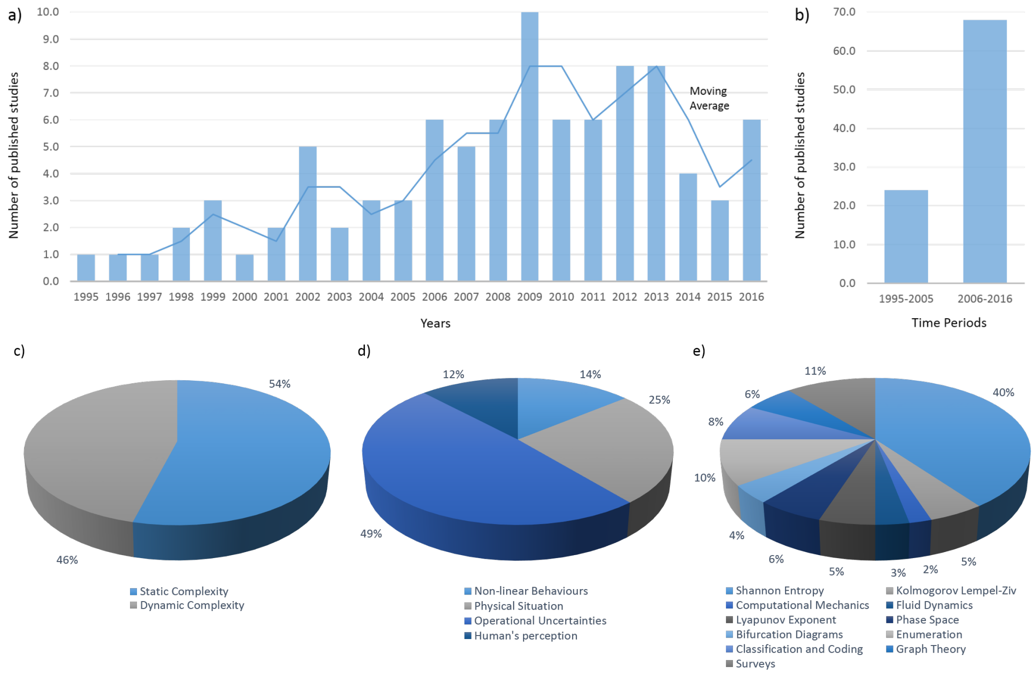

Figure 3.8: Summary of the literature review: *a*) the number of published studies per year (from 1995 to 2016), *b*) the number of published studies between the time periods of 1995-2005 and 1996-2006, *c*) the popularity of complexity type studied: static (0.7312 per study) and dynamic (0.6129 per study), *d*) the popularity of class of symptoms studied: operational uncertainties (0.5376 per study), dynamic behaviours (0.1505 per study), physical situation (0.2688 per study), human perspectives (0.1290 per study) *e*) the popularity of the method employed: Shannon entropy (0.4409 per study), Kolmogorov Lempel-Ziv (0.0538 per study), computational mechanics (0.0215 per study), fluid dynamics (0.0323 per study), maximal Lyapunov exponent (0.0538 per study), phase space reconstruction (0.0645 per study), bifurcation diagrams (0.0430 per study), enumeration (0.1075 per study), classification and coding (0.0860 per study), graph theory (0.0645 per study), surveys (0.1290 per study).

other key stakeholders require practical efficient methods for measurement, which are simply not available to them [Gabriel, 2007a].

It has been observed that more is learnt about a given system during the process of measuring complexity than the analysis of the resulting data [Calinescu et al., 1998]. This demonstrates two shortcomings of complexity measurement: *i*) a disconnect between complexity science and real manufacturing systems engineering means that models are usually unable to evaluate the system at the required level of abstraction, requiring reformulation and thus resulting in non-systemic approaches and *ii*) complexity is measured when the system exists in the physical domain, and thus any measurement or assessment can only have a limited impact for making improvements.

In addition, many researchers have offered measurements that provide a single value of complexity for an entire system. However, a manufacturing system is a combination of multiple sub-systems such as: mechanical, electrical, and control as well as process types including: manual, semi-automatic and fully-automatic [Gullander et al., 2011]. Thus, models that can decompose these concepts systematically and capture the data, can accurately measure the complexity of the system and the sub-systems identifying the source of complexity in the system and thus focused efforts for optimisation.

The classical approach to engineering and design in industry has been heavily reliant either on documents or the expertise of designers, engineers and integrators [Lee et al., 2014]. The paradigm of model-based engineering (MBE) and data driven approaches has emerged in the last decade as a direct result of increased computing power at lower costs. In fact, this is one of the key paradigms of Industry 4.0 [Bloem et al., 2014] although such approaches have been in use before this term was coined. MBE moves the record of authority from documents to digital models allowing engineering teams to more readily understand design change impacts, communicate design intent and analyse a system's design. However, within the context of complexity assessment there remain a number of shortcomings of the existing model-based approach to system's engineering.

Firstly, models are not integrated effectively beyond their respective phase or engineering domain [Kernschmidt and Vogel-Heuser, 2013; Vogel-Heuser et al., 2014]. This has the consequence that there is limited transparency as to what impact change would have outside of a given life-cycle phase resulting in unexpected outcomes, a condition inherent of complex systems. Furthermore, the limited transparency beyond a given engineering domain e.g. electrical vs. mechanical, has a similar consequence.

Secondly, the software tools associated with design and engineering have little to no complexity assessment capabilities within them. As has already been mentioned, this is not used by industry as an indicator to infer that perhaps costs and lead-times may increase, or if indeed the complexity is required, then management and control strategies need to be deployed ahead of time. The first shortcoming identified ties in with the second in that if a given engineering tool software developer was to take the first step in having a complexity assessment tool built in, there would be limited value as it would have trouble translating to adjacent and downstream engineering models. As a consequence of these shortcomings, the complexity measurement process in industry remains tedious, time consuming, and generally non-value adding.

Virtual engineering tools are producing vast amounts of data sets which, if streamlined and integrated, can be used as an input to complexity models. Furthermore, virtual engineering tools, whereby the data structure is extendable, allows additional factors to be modelled as more complexity sources are identified and linked. This approach to complexity measurement has two important benefits over the methodologies presented in the literature: *i*) an assessment of complexity can be made during the design phase so that those designs deemed excessively complex can be flagged and optimised and *ii*) the measurement of complexity is automated and integrated within the virtual engineering tools (or in the case of the cyber-physical systems, data is fed directly from the machine's operation) through to the complexity model resulting in reduced measurement efforts. It is important to note however that fully objective approaches to complexity measurement are not always entirely practical. As a result, approaches such as surveys and questionnaires, while they are susceptible to the subjectivity of those questioned, still offer valuable information and such methods can be improved if they follow a systematic approach.

A clear shortcoming of complexity measurement at the system design phase is a potential lower accuracy due to a lack of operational data, however industry may not necessarily be looking for an exact value. Rather, the requirement is to capture an objective value that is comparable to design alternatives to facilitate in the selection and optimisation of designs. Moreover, a systematic approach to complexity measurement requires bounding, otherwise a complex design cannot be identified. It is therefore of fundamental importance to determine what is acceptable and this requires linking complexity with other key decision criteria such as cost, quality, flexibility and time [Chryssolouris, 2013a]. It is also expected that as engineering tools and methods develop there will be increased scope to introduce mechanisms to measure complexity of manufacturing systems. By considering such parameters, the gap between academia and industry will close resulting in a unified, common understanding of this often misconstrued concept.

3.8 Chapter summary

This chapter has examined the drivers and symptoms of complexity in manufacturing systems and presented a critical review on the analytical and systematic models attempting to analyse and measure manufacturing system complexity. From the review of the literature, the following results has been obtained:

- Complexity in manufacturing systems is a consequence of the evolution of manufacturing firms to adapt today's uncertain manufacturing environment. It is an added-value and provides flexibility and adaptability to manufacturing organisations. However, it also brings the fragility and unpredictability, which will be devastating if complexity is not controlled appropriately.

- Assessment of complexity is an essential requirement of complexity management as it allows manufacturing firms to detect stress points in a manufacturing system and to take most appropriate actions to handle it. The main reason for assessing the complexity is to design and develop engineering systems that are diagnosable, predictable and productive, which leads into reduced costs due to ease of maintenance, foresight and efficient use of resources.

- In the literature, there are many ways to model and measure complexity which have varying pros and cons. As an example, the methods derived from chaos and non-linear dynamics are used to analyse the complexity by means of system's dynamic behaviours which required to be observed in a long time interval, whereas, the heuristics-indices based methods estimate complexity solely based on the system's physical situation but with a low accuracy. The former is used to choose the most appropriate control policy to handle uncertain conditions, while the latter is chiefly employed to compare design alternatives at conceptual stages.

- In recent years, proactive complexity assessment conducting during early design stages has achieved a considerable amount of attention from academia as it enables significant savings in terms of time and cost. However, these measures include either paper based system assessment or face-to-face interviews and questionnaires, for data collection, thus, they are considered as costly and time consuming.

- To fulfil the gap between industry and academia, complexity assessment should include the following features: *i*) it should be able use the data that is reasonably easy to obtain and be able to use that data in a clear and understandable, step-by-step analysis, *ii*) it should be use objective data that can be obtained reliably by multiple observers, *iii*) it should provide the practitioner a tool to compare system designs, *iv*) it must have an intuitive formulation, so that managers can easily recognize what degree of affect that systems changes will have on the measure,*v*) it should permit researchers to quantitatively analyse the relationships between system design and system performance, and *vi*) it should be able to be used in academic research for performing within and across industry research.

The results obtained from this literature review form the guiding principles for the quantitative measures proposed in this research.

Chapter 4

Research methodology

Despite the recent technological advancements, the existing solutions to complexity management are still immature and typically target post-design phases of manufacturing system life-cycle, thus leading to costly and time consuming redesign phases. The presented research focuses on static design complexity of both assembly automation systems and products, and presents two major aims. The former is to present a systematic methodology to quantify static design complexity during early-design stages, and the latter is to achieve a concurrent design evaluation mechanism by integrating the theoretical methodologies into a virtual system design and development tool, where the virtual design data can be streamlined and used as an input to the theoretical models. In this chapter, the static design complexity is formally defined and theoretical origin of the methods used in this research is presented. Moreover, the contribution of the presented study is highlighted and expected outcomes are discussed.

4.1 Definition of complexity

The presented research defines assembly systems as an engineering network consisting of a number of connected components which are working and interacting with each other to realise a common manufacturing goal. The static (structural) complexity of such networks is assumed to be the result of the complexity of individual system elements, and the effects of their connectivity pattern. To formally define the static (structural) complexity, the presented research adopts the following definition proposed by [Sinha and de Weck, 2012] as a base frame which will be used in modelling of static complexity in the subsequent chapters:

"*Static complexity of a network-based system is a function of i) the complexities of individual components, ii) the complexities of pair-wise interactions, and iii) the effects of the system's architectural pattern, which makes the development and management of the system mentally difficult and error-prone.*"

4.2 Origin of the methodology

The theoretical framework presented in this thesis is mainly based on Huckel's molecular orbital theory [Hückel, 1932] which aims to analyse configuration energy of π electrons in conjugated hydrocarbon systems. In Huckel's model, the configuration energy of atomic orbitals is expressed as a function of *i*) self-energy of the individual atoms in isolation, *ii*) interaction energy between interconnecting atoms, and *iii*) the effects of the molecular system topology. In here, the configuration energy outlines the distinctive ability of the interacting system to respond to its surroundings and higher values show an increasing effort required to develop/manage the system [Sinha, 2014].

The Huckel's molecular orbital theory is chiefly introduced to engineering domain by [Sinha, 2014], to analyse complexity of network based engineering systems. In their research, they have argued that any engineering system can be represented by a number of components that are connected to each other in varying ways, where each component can be thought of as an atom and the interfaces between them as inter-atomic interactions (*i.e.* chemical bonds). In this analogy, complexity C associated with the system's inherent structure is defined as below.

$$C = C_1 + C_2 C_3 \tag{4.1}$$

In here, the first term C_1 symbolises the sum of complexities of individual system components, which are designated by α_i:

$$C_1 = \sum_{i=1}^{N} \alpha_i \tag{4.2}$$

This term indicates the technical/ergonomical difficulty/effort associated with the development and management of the component in an isolated condition, and does not require system's architectural information.

The second term C_2 represents the sum of complexities of each pair-wise interaction, which is defined as β_{ij},

$$C_2 = \sum_{i=1}^{N}\sum_{j=1}^{N}\beta_{ij}A_{ij} \tag{4.3}$$

where, A_{ij} defines the binary adjacency matrix representing the connectivity structure of the system:

$$A_{ij} = \begin{cases} 1 & \text{if there is a connection between i and j} \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

Similarly, the term C_2 indicates the technical/ergonomical difficulty/effort associated with the development and management of each pair-wise interaction, and requires knowledge about the inherent nature of each interface as well as the overall system architecture.

The last term, C_3 is a global measure that encapsulates the inherent arrangement of connections and is calculated by the *graph energy* metric (see Nikiforov [2007]).

$$C_3 = \frac{E_A}{N} \tag{4.5}$$

Notice that, the term C_3 requires knowledge of the complete system architecture, and in this sense, contrary to the previous terms, signifies a global effect whose influence could be perceived during the system integration phase Sinha [2014]. Therefore, the term C_2C_3 can be referred as a general indicator of system integration effort. In summary, the analogy defines static complexity of a system (A) in a functional form as follows:

$$C = \sum_{i=1}^{N}\alpha_i + (\sum_{i=1}^{N}\sum_{j=1}^{N}\beta_{ij}A_{ij})(\frac{E_A}{N}) \tag{4.6}$$

Figure 4.1 shows the constituent elements of the complexity metric.

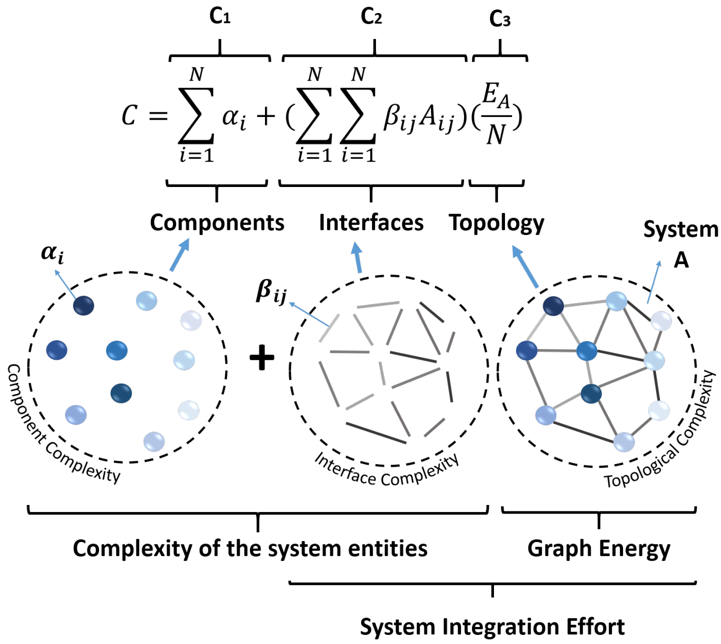

Figure 4.1: Elements of the overall complexity metric.

4.3 Reasons for selection of the analogy

In this research, the Huckel's molecular orbital theory analogy is adopted as the fundemental basis due to a number of reasons.

- First, the approach is objective and mathematically rigorous, and allows us to relate complexity to system development effort in a quantitative and explicit fashion.
- According to Sinha [2014], the mathematical model is valid as it is compliant with Weyuker's criteria [Weyuker, 1988], which provides a set of properties of syntactic complexity measures. Please, see [Sinha, 2014] for more detail.
- The mathematical model has been successfully used to assess complexity of various engineering systems, such as jet engines [Sinha, 2014], and printing systems [Sinha and de Weck, 2013], etc.
- In addition to this, the approach is generic and systemic, and can be adapted and customised for different engineering systems with network like topology.

As an example, the same approach is adapted to assess static complexity of component-based assembly systems in the chapter six.

- The mathematical model is also well-aligned with the component-based design paradigm which is a widely used design methodology in manufacturing systems engineering. This design paradigm is the fundamental approach used in the engineering tool-set called the vueOne virtual manufacturing toolset, within which the mathematical model for measuring complexity of assembly systems will be integrated in the chapter seven.

4.4 The novelty of the research

The presented research is an extension of the methodology proposed by [Sinha, 2014]. The complexity definition proposed in [Sinha, 2014] is further extended by modifying the approach used for calculating the individual elements that contribute to complexity, thererby enabling the approach to be introduced to the domain of assembly. The primary difference between [Sinha, 2014]'s work and the presented study is the calculation of the component and interface complexities where the approach proposed in the latter study is a heuristic-based structured approach that is more applicable to the manufacturing industry. In the following chapter, this novel approach is applied to calculate the assembly complexity of industrial products. Specifically, the proposed work adopts the Design for Assembly (DFA) principles to calculate the component and interface complexities. In chapter 6, the presented approach is adopted to assess the complexity of modular assembly system, where a novel approach is used to calculate the complexity of individual components represented in a system-of-systems representation. The approach uses heuristic criteria scheme to estimate individual components represented in physical and logical domains, which enables a high resolution complexity estimation, especially useful for early design stages. Moreover, the approach is further automated by integrating it with a virtual process planning tool, where various virtual design data can be streamlined as an input to the theoretical model. This enables two important benefits over the approaches proposed in the literature: *i*) an assessment of complexity can be made concurrently with the design phase so that those designs deemed excessively complex can be flagged, modified and optimised, and *ii*) as opposed to the

Table 4.1: Comparison of the previous works on complexity.

	[Sinha, 2014]	[Badrous, 2011b]	Presented study
Complexity definition	A function of component, interface and topological complexity	A function of quantity, diversity and information content	A function of component, interface and topological complexity
Component complexity	Expert opinions	Heuristics	**Heuristics**
Interface complexity	Expert opinions	No	**Heuristics**
Topological complexity	Graph energy	No	Graph energy
Application area	Products (general) Jet engines Printing machines	Assembly products Assembly systems	**Assembly products** **Modular assembly automation systems**
Method of calculation	Pen and paper	Pen and paper	**Automated**
Source of data	Humans and Bill of materials	Bill of materials	**Virtual engineering**

pen-and-paper based methods, the measurement of complexity is automated and integrated within the virtual engineering tools resulting in reduced measurement efforts. **Table 4.1** summarises a comparison of PhD theses published in similar topics, i.e. complexity assessment in manufacturing systems. The text in bold highlights the novelty of the presented study.

4.5 Chapter summary

This chapter briefly explains the origin of the methodology and the originality of the presented research work. In the subsequent chapters, the presented approach is used to assess the complexity of assembly products and systems. The proposed theoretical models are then integrated with a virtual process planning tool to enable concurrent design evaluation of assembly systems during early design phase.

Chapter 5

Complexity of product assembly

In this chapter, a systemic approach is proposed to assess assembly complexity of industrial products. The approach is based on Huckel's molecular orbital theory, and defines complexity as a combination of both the complexity of product entities and their topological connections. In this model, complexity of product entities (*i.e.* components and liaisons) is defined as the degree to which the entity comprises structural characteristics that lead to challenges during handling or fitting operations. The characterisation of entity complexities are carried out based on widely used Design for Assembly (DFA) principles. Moreover, the proposed approach is tested on two case studies from electronics industry for its validity. The results showed that the approach can be used at initial design stages to improve both quality and assemblability of industrial products by reducing their complexity and accompanying risks, thereby helping us to design leaner assembly systems.

5.1 Modelling product assembly complexity

The structure of an assembly product is composed of a set of components and liaisons. Components include: *i*) essential components, *ii*) quasi-components and *iii*) virtual components. Essential components can be individual parts or sub-assemblies that behave as a single unit. Quasi-components are used to connect two essential components. These components include threaded (*e.g.* screws, bolts, nuts, *etc.*) and non-threaded mechanical fasteners (*e.g.* snap fits, rivets, *etc.*). Virtual components, on the other hand, are used to represent non-mechanical fasteners, such as: soldered/welded and glued joints. Liaisons are the interactions that physically attach two components to restraint the motion between them [Lambert and Gupta, 2016]. In general, an assembly task is performed to set up these interactions in sequential order to assemble the final product.

The structure of an assembly product can be represented in multiple ways. One of these, known as liaison diagram, graphically visualises the complete product structure using a non-directed graph. In this representation, components are expressed by nodes, and liaisons are defined by edges. Based on the selected level of detail, liaison diagrams can be illustrated in three different forms: *i*) extended form, *ii*) reduced form and *iii*) minimal form [Vongbunyong and Chen, 2015]. **Figure 5.1** shows an assembly product with five components of which, *A* and *B* are connected by snap-fitting, *B* and *C* are connected with a screw *E*, and *C* and *A* are connected by a weld joint *D*. The extended liaison diagram includes all components, while the reduced form of the liaison diagram representing the product structure more briefly by hiding virtual components and using dashed lines for quasi-components. The minimal form represents the product structure in a more compact way by only including essential components and the direct connections between them in the diagram. This form is the simplest way while keeping the information concerning the essential components visible.

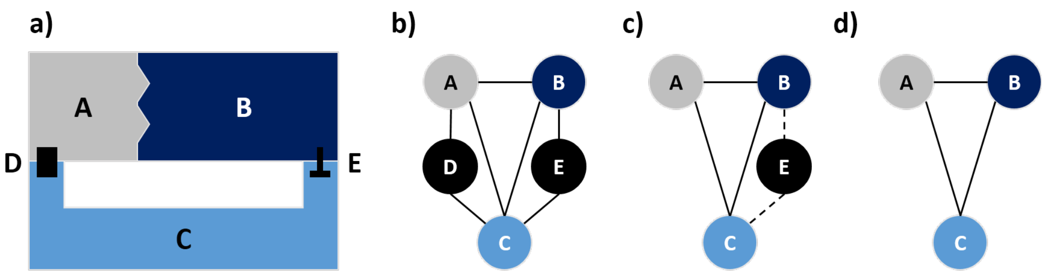

Figure 5.1: Representation of assembly products, *a*) product structure, *b*) extended liaison diagram *c*) reduced liaison diagram *d*) minimal liaison diagram (adapted from [Vongbunyong and Chen, 2015])[Example is taken from the original source, and its liaison diagrams are given in the presented figure].

The assembly product structure can also be represented by the assembly structure matrix (ASM) [Vongbunyong and Chen, 2015]. Unlike design structure matrix (DSM) which visualises dependencies between system components (*e.g.* structural connections, information exchange, material and energy transfers, *etc.*), ASM approach only depicts liaison connections (*i.e.* how components are joined together). The ASM is a N-by-N symmetrical matrix, where each element of the matrix designates the existence of an assembly liaison between two components:

$$[ASM]_{ij} = \begin{cases} 1 & \text{liaison exists between } i \text{ and } j \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

Diagonal elements of ASM are always zero. As an example, the ASM for the extended form of the above-mentioned example is given below.

$$[ASM] = \begin{vmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{vmatrix} \tag{5.2}$$

In this chapter, an assembly product is thought as a stand-alone system consisting of a number of components handled and inserted by either human operators or assembly machines in sequential order to form the product. In here, it is hypothesized that the assemblability of the product is linked to its static complexity, therefore, any reduction in the complexity without compromising product's functionality will enhance the quality of the assembly and reduce associated costs. By adapting the Huckel's approach presented in the previous section, the assembly complexity of manufacturing products C^p is defined as follows:

$$C^p = C_1^p + C_2^p C_3^p \tag{5.3}$$

where, C_1^p, C_2^p, C_3^p represent component, liaison and topological complexity of the product, respectively. This section discusses the rationale behind the estimation of the various elements of the product complexity metric.

5.1.1 Complexity of product components, C_1^p

Components complexity C_1 represents the sum of complexities of individual system components. In case of an assembly product, this term is labelled as C_1^p and calculated as follows:

$$C_1^p = \sum_{i=1}^{N_p} \alpha_i^p \tag{5.4}$$

where, α_i^p represents the complexity of the product component i, and N_p defines the total number of components (excluding virtual components) forming the product. In this context, complexity of a product component is defined as the ergonomical/technical difficulty to interact with the component, and measured based on the degree to which the component has physical characteristics that result in difficulties or problems during its handling during manual and automatic assembly operations. In this research, handling difficulty of assembly components is estimated using a methodology derived from the Lucas Method Chan and Salustri [2005] (**Table 5.1**).

The Lucas Method is a point scale product design analysis method which provides a relative measure of difficulty of both manufacturing and assembly operations. In the approach, issues regarding the handling of assembly components are evaluated by the *handling index*. This index indicates the average handling difficulty of components and it is calculated based on the physical factors of size,

Table 5.1: Complexity of part handling attributes f_h (Source: Chan and Salustri [2005]).

Attribute	Description	f_h
A - Size and weight	Very small - requires handling aids	1.5
(One of the following)	Easy - requires one hand only	1
	Large and/or heavy - requires more than one hand or aid	1.5
	Large and/or heavy- requires hoist or more than one person	2
B - Handling difficulty	Delicate	0.4
(All that apply)	Flexible	0.6
	Sticky	0.5
	Tangible	0.8
	Severely nest	0.7
	Sharp/abrasive	0.3
	Untouchable	0.5
	Gripping problem/slippery	0.2
	Automatic handling - no difficulty	0
C - Alpha Symmetry	Symmetrical - no orientation required	0
(One of the following)	Easy to orient - end to end	0.1
	Difficult to orient - end to end	0.5
D - Beta Symmetry	Rotational orientation is not required	0
(One of the following)	Easy to orient - end to end	0.2
	Difficult to orient - end to end	0.4

weight, handling difficulties and orientation. In this study, the normalised handling index is used to define the complexity of product components α_i^p:

$$\alpha_i^p = \frac{f_h^A + \sum_1^{N_B} f_h^B + f_h^C + f_h^D}{\alpha_{max}^p} \tag{5.5}$$

where, α_i^p is the complexity of i^{th} component, N_B is the number of applicable handling difficulties, and α_{max}^p is the theoretical maximum value for the handling index (6.9). A high value of α_i^p indicates an increased handling difficulty for the corresponding component. Since component complexity C_1^p is a cumulative score, eliminating non-essential components and designing for ease of handling will reduce product's cumulative component complexity.

5.1.2 Complexity of assembly liaisons, C_2^p

The complexity of liaisons C_2^p is the sum of the complexities of pair-wise connections that exist in the product structure. In this study, we only consider connections between essential components as liaisons. Therefore, the calculation of C_2^p is carried out by only considering the minimal form of the product's ASM. By adapting the presented complexity modelling framework, the liaison complexity can be defined as follows:

$$C_2^p = \sum_{i=1}^{N_p^e} \sum_{j=1}^{N_p^e} \beta_{ij}^p ASM_{ij}^{minimal} \tag{5.6}$$

$$[ASM]_{ij} = \begin{cases} 1 & \text{if there is a connection between } i \text{ and } j \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

where, N_p^e is the number of essential components. Complexity in achieving a liaison between essential components i and j (β_{ij}^p) can be expressed by the relationships between the linked components and the nature of the connection. In this study, we adapted the normalised *fitting index* from the Lucas Method to assess individual β_{ij}^p values. The *fitting index* predicts the difficulty of an assembly fitting by penalising the physical attributes that affect the fitting difficulty. These attributes include: the direction of the fitting, insertion type, visibility, *etc*., and is given in **Table 5.2**. complexity of establishing a liaison is calculated as follows:

Table 5.2: Complexity of part fitting attributes f_f (Source: Chan and Salustri [2005]).

Attribute	Description	f_f Manual	f_f Auto
E - Part placing	Self-holding	1	1
(One of the following)	Holding down required	2	1.2
F - Part fastening	Self-securing	1.3	1.1
(One of the following)	Screwing	4	1
	Riveting	4	1.3
	Bending	4	1.6
	Mechanical deformation	4	1
	Soldering or welding	6	1.6
	Adhesive	5	1.2
G - Direction	Straight line from above	0	0
(One of the following)	Straight line not from above	0.1	0.2
	Not straight line and/or bending is required	1.6	1.2
H - Insertion	Single	0	0
(One of the following)	Multiple	0.7	1.2
	Simultaneous multiple insertions	1.2	1.2
I - Restricted vision	Visible	0	0
(One of the following)	Not visible	1	0
J - Difficult to align	No	0	0
(One of the following)	Yes	0.7	0.8
K - Resistance to insertion	No	0	0
(One of the following)	Yes	0.6	0.8

$$\beta_{ij}^{p} = \frac{f_f^E + f_f^F + f_f^G + f_f^H + f_f^I + f_f^J + f_f^K}{\beta_{max}^{p}} \tag{5.8}$$

where, β_{max}^{p} is the theoretical maximum value for the fitting index (12.4). Note that, high β_{ij}^{p} scores indicate an increase in difficulty/effort to achieve the corresponding liaison, which may be eliminated by reducing part insertion difficulties (*e.g.* use of self-secured connections, designing parts with self alignment, increasing visibility, *etc.*).

5.1.3 Complexity of the product's topology, C_3^p

The architectural pattern of a product results in the topological complexity associated with the interactions between components and relies on the combinatorial nature of the system's interconnectivity [Kinsner, 2010]. By following the definition proposed by [Sinha, 2014], topological complexity is expressed as the matrix or graph energy E (see [Nikiforov, 2007]), which is designated by the sum of singular values σ_i of the minimal assembly structure matrix $E_{ASM^{minimal}}$ of the product under consideration.

$$C_3^p = \frac{E_{ASM^{minimal}}}{N_p^e} \tag{5.9}$$

$$E_{ASM^{minimal}} = \sum_{i=1}^{N_p^e} \sigma_i \tag{5.10}$$

This metric outlines the nominal effective dimension entrenched within the connectivity pattern [Sinha, 2014]. According to Sinha Sinha [2014], topological complexity increases as the system's structure shifts from centralised architectures to more distributed architectures. Furthermore, topological complexity is divided into three regions: $C_3 < 1$ hypoenergetic (centralised architecture), $1 \leq C_3 < 2$ transitional (hierarchical/layered architecture), and $C_3 \geq 2$ hyperenergetic (distributed architecture) [Sinha, 2014]. In a practical manner, topological complexity indicates the '*intricateness*' of structural dependency among assembly components [Sinha et al., 2017]. Topological complexity C_3^p allows us to differentiate the product structures with similar component and liaison complexities, and to better predict the integration effort.

5.2 Empirical validation

In this section, empirical validation of the presented approach is carried out by a series of simple experiments. In these experiments, participants were asked to assemble organic molecule structures from a molecular modelling kit based on a clear 2D assembly work instruction. The assembly time of each molecule is recorded, and a statistical model correlating the assembly complexity of the molecule models to their average assembly time is developed.

5.2.1 Materials

In the study, eight different ball-and-stick molecule structures with a reasonable spectrum of assembly complexity, are selected to be assembled by the participants. All ball-and-stick models are based on molecular structures that could be built from the available molecular tool kit and given in **Figure 5.2**.The models include hydrogen (white), carbon (black), oxygen (red), nitrogen (blue) and phosphorus (green) atoms and three kinds of chemical bonds, i.e. short single connectors (compact single covalent bonds), medium connectors (single covalent bonds), and long flexible connectors (double and triple covalent bonds). As an example, the assembly schematic of the model number 8 is given in **Figure 5.3**. This molecule structure consists of 35 atoms, 6 flexible long connectors, 16 medium connectors, and 16 compact single connectors and has a chain type centralised internal topology ($E_{ASM}/n = 1.22$).

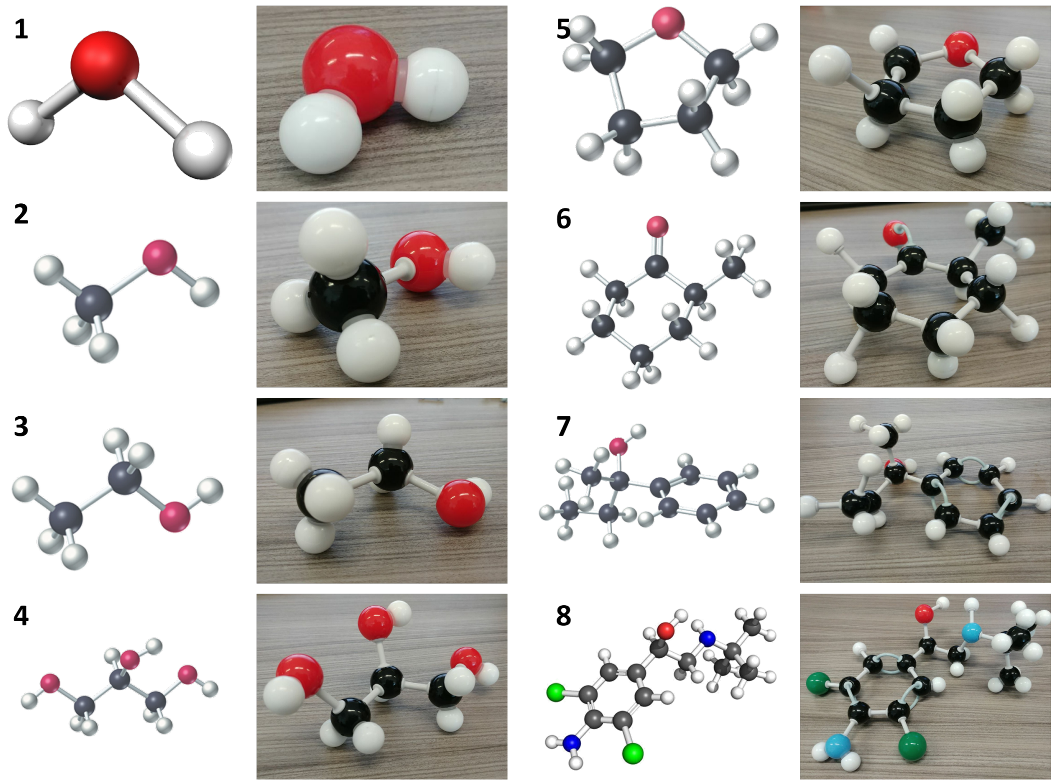

Figure 5.2: The eight molecule ball and stick models used in molecule assembly experiments

It is assumed that all atoms and connectors are an essential component, where liaisons between them are achieved by a non-mechanical fastening method. Based on the proposed methodology, component and liaison complexities are calculated as given in **Table 5.3**. Assembly complexity of each molecule structure is

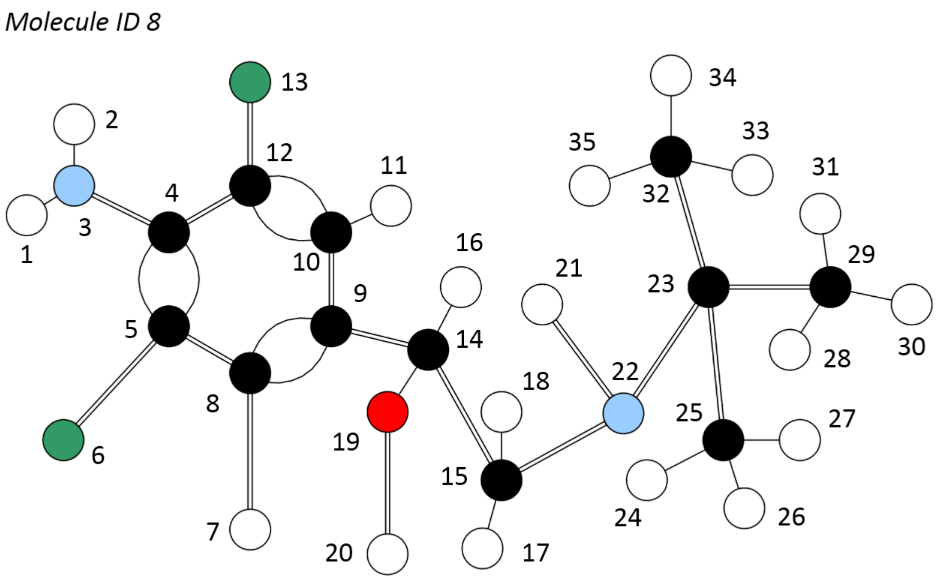

Figure 5.3: The assembly schematics of ball and stick model number 8. In schematic, atoms are coded with colours; and the connectors are represented as: single edges (short single connectors), double edges (medium connectors) and curved edges (long flexible connectors).

calculated as follow.

$$C = (m_a\alpha_a + m_c\alpha_c + m_m\alpha_m + m_f\alpha_f) + n\beta\frac{E_{ASM}}{m_a + m_c + m_m + m_f} \tag{5.11}$$

where, n is the total number of liaisons, m_a is the total number of atoms, m_c, m_m and m_f are the total number of compact, medium and long flexible connectors, respectively. **Table 5.4** shows the complexity scores of eight molecule ball-and-stick model used in the experiments.

Table 5.3: Component and interface complexities.

Entity	Complexity
Atom (Any)	0.145
Compact single connector	0.174
Medium connector	0.174
Long flexible connector	0.261
Non-mechanical fastening	0.186

Table 5.4: The structural complexity results of eight molecule ball-and-stick models.

ID	n	m_a	m_c	m_m	m_f	C_1	C_2	C_3	C_{23}	C
1	4	3	2	0	0	0.78	0.74	0.94	0.70	1.48
2	10	6	4	1	0	1.74	1.86	0.97	1.80	3.54
3	16	9	6	2	0	2.70	2.98	1.00	2.98	5.67
4	26	14	8	5	0	4.29	4.84	1.11	5.37	9.66
5	26	13	7	6	0	4.15	4.84	1.03	4.98	9.13
6	42	20	8	11	2	6.73	7.81	1.06	8.28	15.01
7	56	25	9	13	6	9.02	10.42	1.12	11.67	20.68
8	76	35	16	16	6	12.20	14.14	1.22	17.25	29.45

5.2.2 Procedures and participants

In order to explore the correlation between assembly complexity of molecule models and their average assembly time, a series of experiments were conducted with the participation of human volunteers. The participants received an initial briefing and they were shown the test set-up to familiarise themselves with the atoms and chemical bonds provided by the molecule tool kit. The participants were given the completely unassembled kit and a clear work-instruction for each assembly. The experiments were carried out by eleven participants with similar backgrounds (e.g. PhD students and research fellows). The participants were asked to assemble molecule structures as quickly as possible without any assembly defect. Any defect during the assembly process involves a rework increasing the total assembly time. During the experiments, the total assembly time is recorded as below.

$$T_{assembly} = T_{perception} + T_{mentaldecision} + T_{actionexecution} + T_{rework} \tag{5.12}$$

Please note that, only total assembly time was tracked without the constituent time elements, and the assembly structure was disassembled on completion following which the next work instruction was shown to the participant under study.

5.2.3 Design of experiments

In these experiments, the total assembly time of each ball-and-stick structure model presented in the previous section were tracked and considered as their development effort/cost. **Table 5.5** shows the response model of each assembly: the average, minimum, and maximum assembly times and the standard deviations.

Table 5.5: The results of molecule assembly experiments (m is the total number of essential components).

ID	n	m	E_{ASM}/m	C	Assembly time (s)	Minimum time (sec)	Maximum time (sec)	σ(sec)
1	4	5	0.94	1.48	16.24	11.54	23.47	4.42
2	10	11	0.97	3.54	32.82	26.55	38.86	4.44
3	16	17	1.00	5.67	50.03	38.82	68.04	10.55
4	26	27	1.11	9.66	91.65	71.47	108.48	12.89
5	26	26	1.03	9.13	87.15	74.65	100.59	8.31
6	42	41	1.06	15.01	181.19	148.98	220.45	24.53
7	56	53	1.12	20.68	236.32	201.06	302.59	35.45
8	76	73	1.08	29.45	323.25	240.18	344.12	38.01

According to the result, the individual variance is small for lower static complexity, since it is easier for humans to see the best way of assembling less complex systems; errors and reworks are unlikely and the time to understand and process the information is small. For more complex structures, time for cognitive processing and likely rework becomes larger and can lead to a larger variance between participants. It is shown that the standard deviation increases with the static complexity.

5.2.4 The relationship between complexity and system development effort

A linear regression model is used to analyse the relationship between complexity and system development effort. This model is selected due to the small sample size (n=8). However, more experiments are required to determine the actual trend of the relationship. In this model, product complexity measured by the proposed approach, is used as the predictor of the system development effort (i.e. assembly time), with 95 percent of confidence level:

$$Y = A + BX \tag{5.13}$$

where, Y is the estimated product development effort, X is product assembly complexity, and A and B are the model parameters that are estimated by the least squares method (see for further explanation [Stigler, 1981]). All computation were performed in Minitab$^{\text{TM}}$ environment. The final single variable parametric model relating system development effort (Y) to complexity (X) for the data set achieved

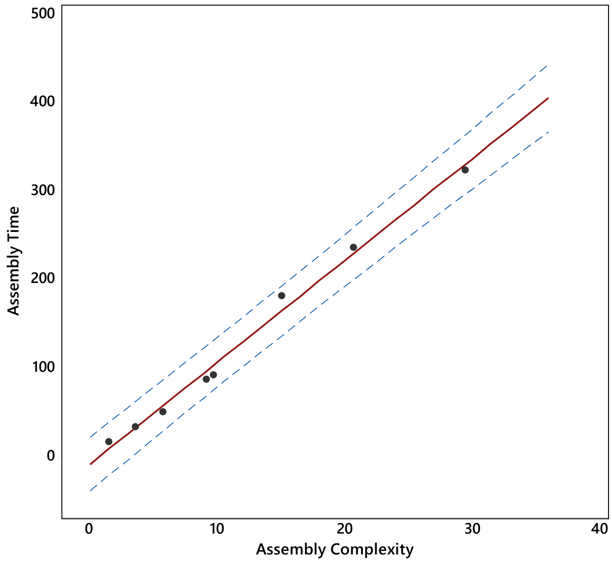

Figure 5.4: Regression plot of single variable parametric model (Dashed lines represent upper and lower confidence intervals).

from the ball and stick model assembly experiments, is given by:

$$Y = -8.689 + 11.50X \tag{5.14}$$

Accordingly, the relationship between assembly time and product assembly complexity is found to be statistically significant ($p < 0.05$) The value of the correlation coefficient, R, is found as .9902. Regression plot and model quality parameters are given in **Figure 5.4** and **Table 5.6**, respectively. The regression analysis showed that system development effort increases with increasing complexity, which is accompanied with an increase in the difficulty, mental and physical exertion and possibility of human errors. It is also observed that the variations in assembly time increases with the degree of complexity. This is essentially related with the capability of humans to manage increasing complexity.

Table 5.6: Model parameters, and model quality measures.

Model	a	b	RSquare
$a+bX$	-8.689	11.50	0.9916

5.3 Case studies

The combination of the complexity elements allows us to comprehend how the structural characteristics of a product impact the complexity of its assembly process. This section presents the demonstration of the proposed metric on real engineering products.

5.3.1 Printed circuit board (PCB) pressure recorder device

The case presented in this section is of the assembly of a pressure recorder device. The product data is taken from the DFA handbook [Boothroyd and Dewhurst, 1987]. **Figure 5.5** shows the original design of the pressure recorder assembly and its liaison diagram. The assembly consists of six essential and eleven quasi-components, and eight liaisons. The analysis results of component and liaison

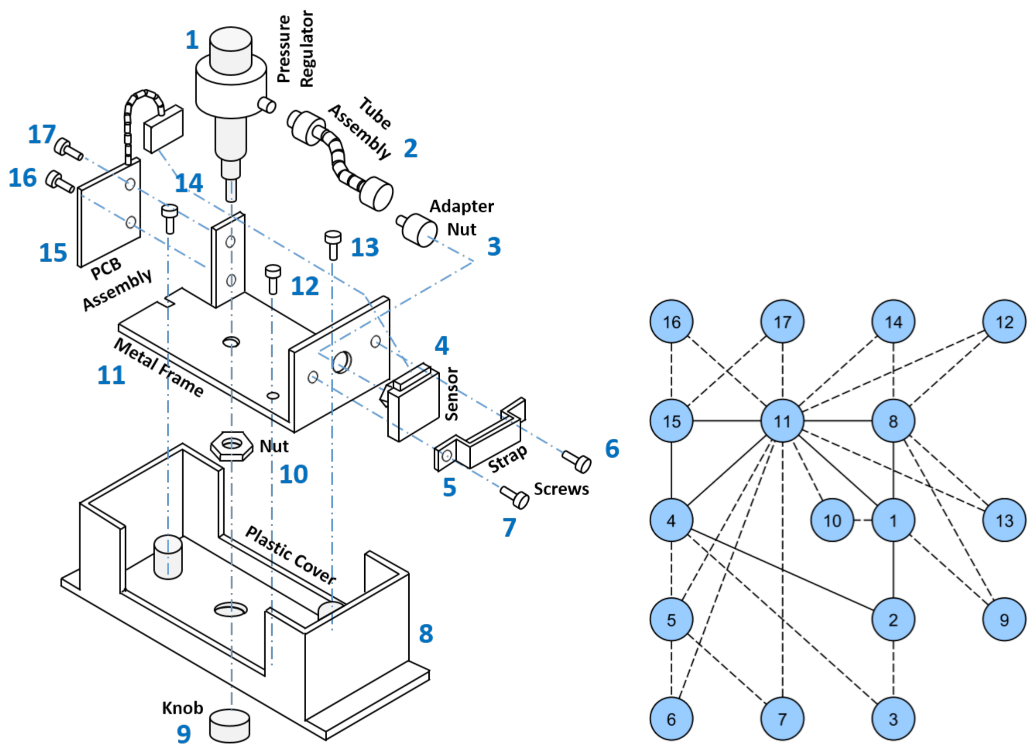

Figure 5.5: Initial design of the pressure recorder device and its liaison diagram $C^p = 7.776$ $C_3^p = 1.210$. (adapted from [Boothroyd and Dewhurst, 1987] [Product schematic is taken from the original source, whereas the liaison diagram is a unique contribution].)

complexities are shown in **Table 5.7** and **Table 5.8**, respectively. Complexity of the product's topology is recorded as 1.396 indicating a hierarchical architecture. According to the results, the overall complexity of the product's assembly C^p is calculated as 7.776.

As a next step, the original pressure recorder is re-designed based on the design for serviceability (DFS) principles (see [Dewhurst and Abbatiello, 1996]), as it is shown in **Figure 5.6**. In the improved design, the component number eleven of the initial design is completely removed, as it is tightly coupled with the remaining structure, and the component structure is re-arranged to accommodate the fewest possible number of quasi-components. **Tables 5.9** and **5.10** show the component and liaison complexities of the improved pressure recorder design, respectively. The topological complexity of the new design is noted as 1. These values indicate that the improved design has 63.2% reduction in the overall product complexity when compared to the original design (from 7.776 to 2.864).

Figure 5.7 shows the graphical comparison between complexity scores of

Table 5.7: Calculation of component complexities - original pressure recorder design.

	Component																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
f_h^A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\sum f_h^B$	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0
f_h^C	0.1	0	0.1	0.1	0.1	0.1	0.1	0.5	0	0	0.5	0.1	0.1	0.1	0.1	0.1	0.1
f_h^D	0.2	0	0	0.2	0.2	0	0	0.2	0	0	0.4	0	0	0	0.2	0	0
α_i^P	0.19	0.23	0.16	0.19	0.19	0.16	0.16	0.25	0.14	0.14	0.28	0.16	0.16	0.16	0.28	0.16	0.16
C_1^p																	**3.159**

Table 5.8: Calculation of liaison complexities - original pressure recorder design.

	Liaison							
	1-2	1-8	1-11	2-4	4-11	4-15	8-11	11-15
f_f^E	2	1	1	2	2	2	2	1
f_f^F	1.3	4	4	4	4	1.3	4	4
f_f^G	0.1	0	0	0.1	0.1	0.1	0	0.1
f_f^H	0	0	0	0	0.7	0	1.2	0.7
f_f^I	0	1	1	0	0	0	0	0
f_f^J	0	0	0	0.7	0	0.7	0	0
f_f^K	0	0	0	0	0.6	0	0.6	0
β_{ij}^P	0.27	0.48	0.48	0.55	0.60	0.33	0.63	0.47
C_2^p								**3.815**

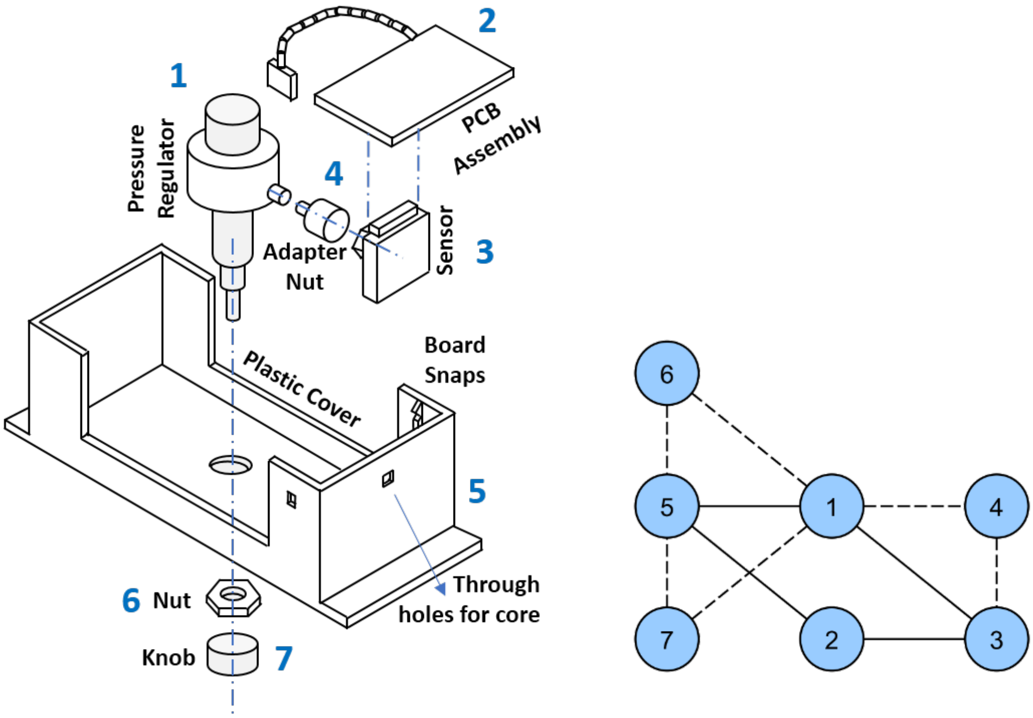

Figure 5.6: Redesign of the pressure recorder device and its liaison diagram C^p = 2.864 C_3^p = 1.000. (adapted from [Boothroyd and Dewhurst, 1987][Product schematic is taken from the original source, whereas the liaison diagram is a unique contribution].)

Table 5.9: Calculation of component complexities - improved pressure recorder design.

	Component						
	1	2	3	4	5	6	7
f_h^A	1	1	1	1	1	1	1
$\sum f_h^B$	0	0.6	0	0	0	0	0
f_h^C	0.1	0.1	0.1	0.1	0.5	0	0
f_h^D	0.2	0.2	0.2	0	0.2	0	0
α_i^P	0.19	0.28	0.19	0.16	0.25	0.14	0.14
C_1^p							**1.348**

the analysed pressure recorder designs. The presented complexity model indicates that the C^p score of initial design is 63.2% higher than that of the improved design. Since, the improved design uses only three quasi-components and a snap-fit, the liaison complexity is reduced by 60.3%. Moreover, the contribution of component complexities has been reduced by 57.3% in the improved version through the elimination of non-essential fasteners. Additionally, the improved version also indicates

Table 5.10: Calculation of liaison complexities - improved pressure recorder design.

	Liaison			
	1-3	1-5	2-3	2-5
f_f^E	2	1	2	2
f_f^F	4	4	1.3	1.3
f_f^G	0.1	0	0	0.1
f_f^H	0	0	0	0
f_f^I	0	0	0	1
f_f^J	0	0	0	0
f_f^K	0	0	0	0
β_{ij}^p	0.49	0.40	0.27	0.35
C_2^p				**1.516**

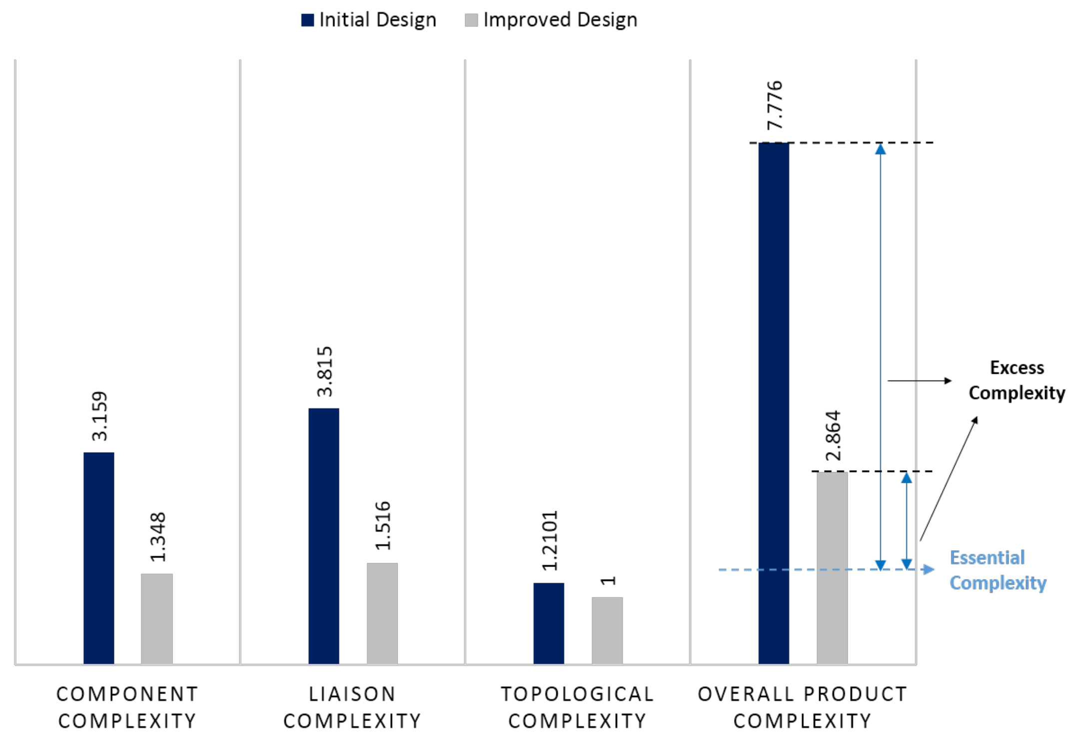

Figure 5.7: Comparison between complexities of initial and improved pressure recorder designs. (Please note that, the value of essential complexity is arbitrarily selected.)

a 17.4% reduction in the topological complexity score. As expected, the changes to the design have enhanced the handling and fitting attributes of the components, while increasing simplicity in the product's assembly topology. This reduces the excess complexity which is the difference between actual product complexity and the

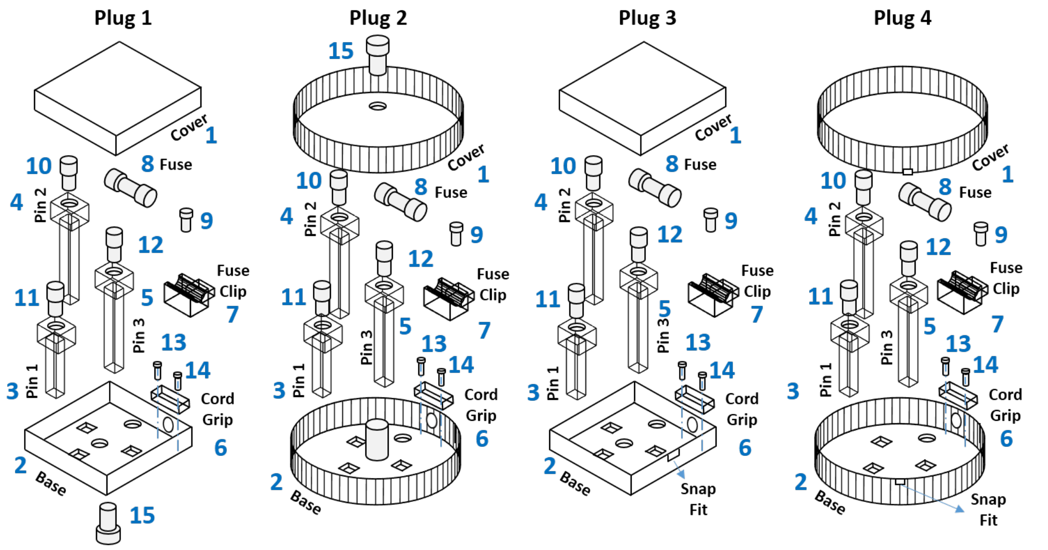

Figure 5.8: Four variations of a three-pin power plug assembly (Source: [Samy and El-Maraghy, 2010b].)

essential complexity that is the non-measurable basic level of complexity required by the product to satisfy its functional requirements. The results demonstrate that the proposed approach has accurately highlighted the effects of design improvements on assembly complexity in an explicit fashion.

5.3.2 Three-pin electric power plugs

The second case is taken from [Samy and ElMaraghy, 2010b], and is of the assembly of four three-pin power plugs (**Figure 5.8**) which are members of a product family. The variants consist of a number of similar components including the cord grip, fuse, fuse clip, pins, *etc*., and are handled by the same fixture as the four plug variants have identical base designs. The main difference between the variants is that the variants 1 and 2 use a direct screw to assemble the base and the cover components together, while the variants 3 and 4 use snap-fits to realise this connection. Moreover, the screw connecting the base and cover components is inserted from below in the first variant and from above in the second variant. In this section, assembly complexities of these variants are analysed to test the sensitivity of the proposed approach and the results are compared against the results found on the literature.

Topological complexity of the product is recorded as 0.847 for all variants

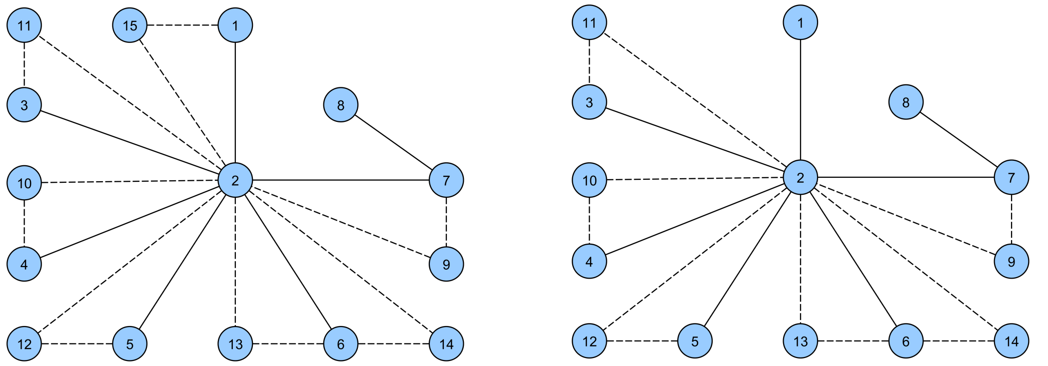

Figure 5.9: Liaison diagram of the three-pin plug variants, *left*: Plugs one and two, *right*: Plugs three and four, $C_3^P = 0.847$

(**Figure 5.9**). This value highlights that the product has a centralised architecture. All plugs are analysed, and overall product assembly complexities are calculated as shown in **Table 5.11**. According to the results, the plug variant one is found as

Table 5.11: Calculation of product assembly complexities - All variants.

	Component Complexity					Liaison Complexity					Overall Complexity			
α_i^P	1	2	3	4	β_{ij}^P	1	2	3	4		1	2	3	4
1	.19	.16	.19	.16	1-2	.55	.54	.19	.19	C_1^P	2.739	2.681	2.580	2.522
2	.19	.16	.19	.16	2-3	.46	.46	.46	.46	C_2^P	3.250	3.242	2.877	2.877
3	.19	.19	.19	.19	2-4	.46	.46	.46	.46	C_3^P	0.847	0.847	0.847	0.847
4	.19	.19	.19	.19	2-5	.46	.46	.46	.46					
5	.19	.19	.19	.19	2-6	.54	.54	.54	.54					
6	.19	.19	.19	.19	2-7	.60	.60	.60	.60					
7	.28	.28	.28	.28	7-8	.19	.19	.19	.19					
8	.22	.22	.22	.22										
9	.16	.16	.16	.16										
10	.16	.16	.16	.16										
11	.16	.16	.16	.16										
12	.16	.16	.16	.16										
13	.16	.16	.16	.16										
14	.16	.16	.16	.16										
15	.16	.16												
C_1^P	2.739	2.681	2.580	2.522	C_2^P	3.250	3.242	2.877	2.877	C^P	5.491	5.426	5.025	4.967

the most complex design. **Tables 5.12** and **5.13** show the component and liaisons complexity results of the plug variant one. **Figure 5.10** illustrates the product complexity results for all variants. Even though the differences between complexity scores are very small, these differences are still traceable. The variant one has a higher cumulative component complexity (2.739) than the other three plugs, as its base and cover have more asymmetric shapes. On the other hand, the plug variants

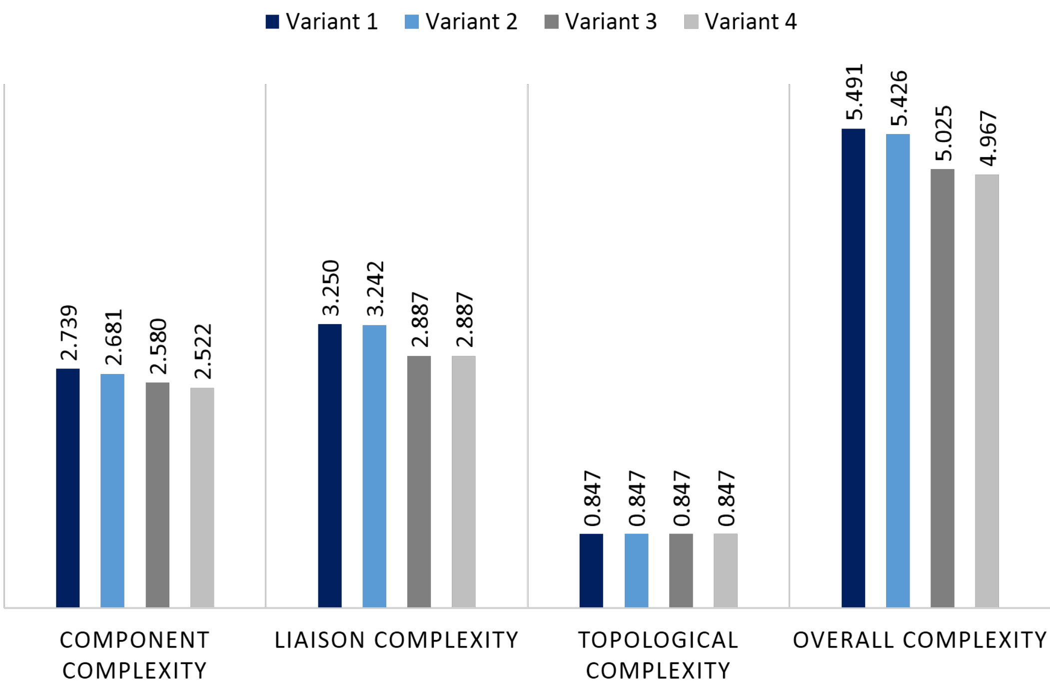

Figure 5.10: Product complexity result for all three-pin plug variants.

Table 5.12: Calculation of component complexities - Plug variant 1.

	Component														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
f_h^A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\sum f_h^B$	0	0	0	0	0	0	0	0.4	0	0	0	0	0	0	0
f_h^C	0.1	0.1	0.1	0.1	0.1	0.1	0.5	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
f_h^D	0.2	0.2	0.2	0.2	0.2	0.2	0.4	0	0	0	0	0	0	0	0
α_i^P	0.19	0.19	0.19	0.19	0.19	0.19	0.28	0.22	0.16	0.16	0.16	0.16	0.16	0.16	0.16
C_1^P															**2.739**

one and two require an additional screw to complete the liaison one, which slightly increases their cumulative component complexity scores. Moreover, it has been recorded that the variants one and two have higher cumulative liaison complexity scores than the variants three and four, as they use mechanical fastening method instead of snap-fits to achieve liaison one. This shows that the effects of changing structural attributes on the product assembly complexity are successfully tracked using the proposed approach.

The calculated complexity results are also compared with the estimations proposed by [Samy and ElMaraghy, 2010b] for same product variants (**Table 5.14**). In their study, complexity of assembly products is calculated using a heuristic

Table 5.13: Calculation of liaison complexities - Plug variant 1.

	Liaison						
	1-2	2-3	2-4	2-5	2-6	2-7	7-8
f_f^E	2	1	1	1	2	2	1
f_f^F	4	4	4	4	4	4	1.3
f_f^G	0.1	0	0	0	0	0	0
f_f^H	0	0	0	0	0	0.7	0
f_f^I	0	0	0	0	0	0	0
f_f^J	0.7	0.7	0.7	0.7	0.7	0.7	0
f_f^K	0	0	0	0	0	0	0
β_{ij}^p	0.55	0.46	0.46	0.46	0.54	0.60	0.19
C_2^p							**3.250**

Table 5.14: Comparison between product complexity and total assembly time.

	Product complexity		
Variant	Presented study	Samy and ElMaraghy [2010b]	Time (secs)
Plug 1	5.49	5.74	38.66
Plug 2	5.43	5.70	37.02
Plug 3	5.02	4.72	31.16
Plug 4	4.97	4.70	29.52

methodology, in which the complexity is defined as a combination of quantity, diversity and the content of the information. According to the comparison, a similar trend has been observed in the estimations proposed by the presented study and [Samy and ElMaraghy, 2010b]. **Figure 5.11** shows the correlation between the calculated product assembly complexity and the approximate assembly times derived from the DFMA analysis (the data is taken from [Samy and ElMaraghy, 2010b]). According to the results, a strong positive correlation is found between the product assembly complexity calculated by the proposed approach and assembly time of the variants derived from the DFMA (see [Boothroyd and Alting, 1992]) ($R^2 = 0.9918$, a linear fit is used under a 95% confidence interval, Assembly Time (sec) = -51.20305 + 16.321635*Complexity). The results show that assembly time increases with an increase in the complexity. This is in consensus with the earlier hypothesis, and accordingly, the increased product complexity demands extra effort from the operators, thereby increasing the assembly time.

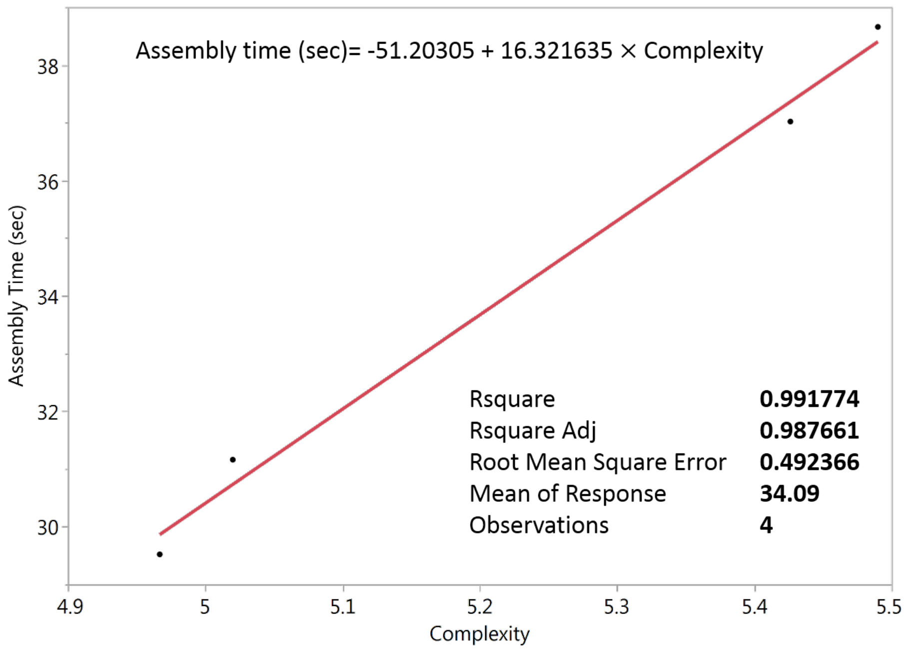

Figure 5.11: Correlation between assembly time and product complexity for three-pin power plug variants.

5.4 Chapter summary

In this chapter, a systemic approach to measure assembly complexity of manufacturing products has been proposed that allows the designer to track the root causes of complexity in the initial design stages. Accordingly, the component and liaison complexity are measured along with the novel methodology to assess the topological complexity of the product architecture. Moreover, the approach is based on a scientifically validated empirical model and tested on two assembly cases from electronics industry. The results are in accordance with the proposed hypothesis and the variation in the product assembly complexity for the different cases, based on the product design is validated. The proposed approach solely requires physical design information and thus, can be considered as practical, especially for initial design stages, than the approaches requiring real production data. The approach can also be extended to include both process sequence and workspace related elements, such as: the type of part presentation, tool changes, *etc.*, for optimising products' assembly sequences.

Chapter 6

Complexity of assembly automation systems

Assembly automation has become a challenging task in the last decades, due to the high-variety production induced by product evolution and mass customisation. Although increased variety enables manufacturers to satisfy a broad range of customer requirements, it is a major contributing factor to increased system complexity, which is generally believed to be one of the primary causes of the present difficulties in the manufacturing domain. As previously mentioned, one key solution to mitigate negative impacts of complexity is its early stage assessment, which can help designers to rationalise system designs, and further compare various design alternatives that meet functional requirements. In this chapter, the complexity modelling analogy presented in the chapter four, is revisited with a set of novel extensions to assess static complexity of component-based assembly system designs. The approach defines assembly systems as a constellation of basic components which can be represented either in physical or logical domains. Accordingly, static complexity is expressed as the combination of complexity of both system's components and their interconnectivity resulting from the integration of such constellations in a multi layered network. The proposed approach is demonstrated on the Festo modular production system (MPS) for its validity. The results are analysed in the light of the opinions of manufacturing experts. Accordingly, it is concluded that the approach is useful in designing productive and controllable systems through increasing their predictability by prioritising and then reducing complexity of critical areas and/or selecting the optimal system configurations among various design alternatives. Moreover, to overcome its disadvantages at large scale design projects, the integration of the methodology with virtual engineering is suggested as a potential solution.

6.1 Early life-cycle phase

During the early life-cycle phase of the manufacturing systems, the main aim is to identify the overall structure of the system, through the decomposition of its functions into sub-functions [Chmarra et al., 2008], and through finding the suitable physical components that can realise corresponding sub-functions [Pahl and Beitz, 2013]. In this context, overall architecture of the system includes not only geometric information, but also non-geometric phenomena such as control architecture and its relations to the overall system architecture [Komoto and Tomiyama, 2012]. According to the V-Model of system development, conceptual design phase is called as "*system architecting*", in which the system requirements are identified, and distributed into subsystems and further components [Komoto and Tomiyama, 2012]. At the lowest layer of decomposition, all sub functions should be realised by essential entities called as "*components*" [Pahl and Beitz, 2013]. According to Komoto and Tomiyama [2012], "*components are called as machine elements, established components, and mechanisms in mechanical design, and fundamental building blocks i.e. state transitions diagrams that represent sensors, actuators, and controller behaviours, in control design*". Once the building blocks of the system are obtained, decomposition of the system builds around finding for relevant components that can realise the decomposed sub-function in a specific configuration (i.e. embodiment) [Komoto and Tomiyama, 2012]. In the last stage of the system architecting, these components are integrated, validated, and verified [Clarkson and Eckert, 2010].

6.2 Architectural modelling of component-based assembly systems

Modelling a complex system which represents simplified formation analogous to the original, is one of the key requirements to better identify and assess system complexity. Component-based assembly systems are one of the leading approaches to the effective management of the system reconfiguration [Lee et al., 2007]. Component-based assembly systems are composed of a number of components each performing a specific functionality of the assembly processes (fitting, handling, transportation, feeding, *etc*.) [Chryssolouris, 2013b]. These systems are typical examples of complex systems, which are consisted of a set of physical parts which are regulated by means of a control system.

In component-based assembly systems there are two types of components, physical components (i.e. equipment) and logical (i.e. software) components. Therefore, based on a unified object-oriented modelling language proposed by [Secchi et al., 2007], conceptual architecture of a component-based assembly system is defined by a two-layered system-of-systems representation (**Fig. 6.1**). This representation is proposed due to the nature of cyber-physical systems, and can be extended to include other industrial IT and communication layers, as the design proceeds into the later stages. Please note that, the system-of-systems notion used here, is also satisfy the architectural principles proposed by [Maier, 1996].

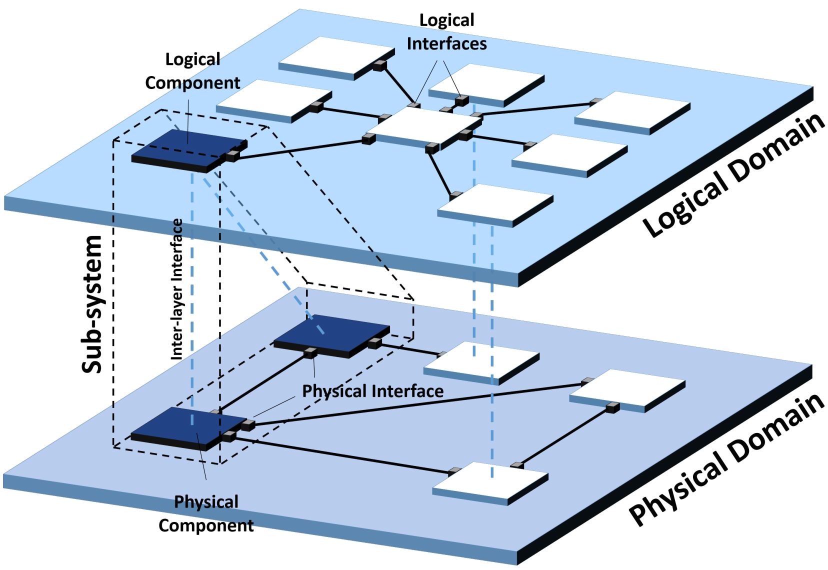

Figure 6.1: Two layered system-of-systems definition of component-based assembly systems (conceptual design stage).

In the presented model, the logical domain (resembling the ISA-95 level 2a and above (see [Commission et al., 2003])) defines management and control policies needed to govern the states and behaviours of physical devices. The physical domain (resembling the ISA-95 levels 0 and 1 (see [Commission et al., 2003])), on the other hand, describes the physical structure of the system, and represents actuation and static functions to physical processes within the automation system. Please note that, the presented study only focuses on the levels of 0-1-2 of ISA-95-related enterprise architectures.

A component, in this context, is a basic unit of the system which at a finer level may compose of a set of indecomposable elements, and capable of functioning either autonomously and/or integrated with other components to perform the desired functions [Chinnathai et al., 2017]. Components can facilitate at least one active/passive function towards the completion of assembly processes. As an example, a robot manipulator can able to provide an active functionality to the system (e.g. transfer parts within and across stations), whereas, a passive fixture can constraint motions which is required to hold a product. Components have standardised interfaces and explicit dependencies which can be deployed independently and are subjected to compositions to build an assembly automation system [Lee et al., 2005]. These interactions can be observed within and between above-mentioned layers. The integration of system components is performed through the combination of the physical components resulting in a physical architecture of the system and through the integration of the logical components resulting in a logical (control) system architecture. The integration of these two layers (i.e. information mirroring) results in a final system architecture, where the system behaviours can be realised in a controlled and synchronised manner.

6.2.1 Physical system design

The physical design of a component-based assembly system is composed of a set of physical equipment connected to each other through a number of physical interfaces. In here, a physical component is the core constituent of the system, which has to be installed and commissioned as a part of the system development phase. The literature defines the following as core components of an assembly system [Ahmad et al., 2016a; Bi et al., 2007; Farid and McFarlane, 2007]:

- Mechanisms required to transfer parts within and across assembly stations. These components often have a flexible level of reachability and can quickly adapt to changes in positional requirements,

- Holding components, i.e. jigs, fixtures and clamps. These equipment are used during assembly processes and part transport and designed for a part/product family with flexible features to support alignment and holding,

- Buffering and storage components which are required to hold parts prior to being introduced into the system that have positional variability,

- Feeding components which are used to transfer parts from buffers to be processed that have positional variability,

- Work holders, e.g. grippers, to handle parts that have changeable functionality due to inherent modularity and that efficiently integrate with the moving mechanism.

6.2.2 Logical system design

Logical design of an assembly automation system can be thought as an object-oriented software system consisted of a set of integrated software components regulating the operations of field devices in a synchronised and controlled manner. These components are used for high-level control activities, such as: event-driven execution control, error-handling, and planning and scheduling, *etc*, and may communicate through event-driven protocols. The behaviour of each component in the logical layer can be described through various approaches including: finite state machine (FSMs), UML state charts, petri nets or Gantt Charts. According to Dai and Vyatkin [2013], the IEC 61131-3 standard [Karl-Heinz John, 2003] used for the design of Programmable Logic Controller (PLC) architectures is one of the most common approaches in industrial automation. In this standard, programing organisation units (POU) (i.e. functions and function blocks (FBs)) are often referred as reusable logical components. The IEC 61131-3 standards FB encapsulates a certain functionality and can be linked to other FBs through input and output interfaces (**Figure 6.2a**). In this standard, each FB contains one algorithm which is written in one of any IEC 61131-3 languages [Zoitl and Vyatkin, 2009]. Another widely used standard in manufacturing industry, IEC 61499, includes the event driven FBs, which are invoked only when an event arrives to one of their event inputs (**Figure 6.2b**). During rest of the operation the FB remains passive. The most important

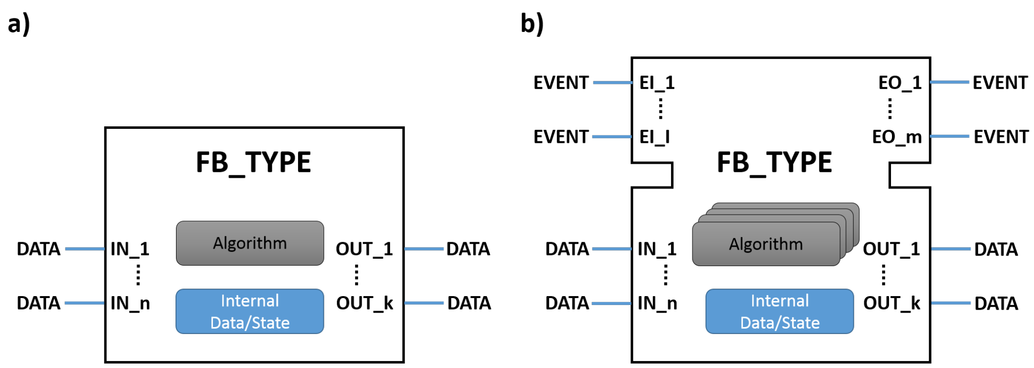

Figure 6.2: a) The IEC 61131-3 and b) the IEC 61499 FB architectures (Source: [Zoitl and Vyatkin, 2009].

aspect introduced by the IEC 61499 [Zoitl and Vyatkin, 2009] is the event interface. Moreover, FBs in the IEC 61499 may contain different algorithms which neither visible nor accessible from the outside. Physical and logical components can be mapped by **electrical dynamic** interfaces. This interface type represents the data transfer between logical and physical components, and can be counted as a directional relationship. Moreover, this relationship does not necessarily to be one-to-one mapping; i.e. the logical behaviour of a physical component can be defined using multiple logical components or multiple physical components can be controlled by a single logical component, *etc*.

6.3 Formulation of system DSM

The design description of manufacturing systems can be represented through various ways. One of the most common approaches, 'Design Structure Matrix' (DSM) is a compact way to visually represent system structures in a square matrix. According to [Browning, 2001], "*DSM is the equivalent of an adjacency matrix in graph theory, and is used in systems engineering and project management to model the structure of complex systems or processes, in order to perform system analysis, project planning and organization design, etc*". A DSM depicts the overall system architecture by visualising the relationships between its constituent components, such as: mechanical connections, spatial interactions, and dependency patterns, etc. [Eppinger, 2005]. Such model analyses system decomposition into subsystems, and further components. The concept of DSM is further expanded by [Maurer, 2007] to analyse systems with multiple domains, each having multiple components, connected by various relationship types: the Multiple Domain Matrix (MDM).

According to the high-definition design structure matrix approach proposed by [Tilstra et al., 2012], the interactions between system components can be defined as:

- **Structural steady state** represents contacts between two physical components where they impose a steady state mechanical load on each other. This is a symmetrical relationship.

- **Structural dynamic state** defines the fluctuating force or displacement between two physical components. This can be a directional relationship.

- **Spatial** defines the relationship between two physical components when they are touching each other or when adjacency and orientation are important between them. This is a symmetrical relationship.

- **Energy** describes energy transfer/exchange between two physical components. This can be a directional relationship.

- **Information** states information exchange between two physical components. This is often a symmetrical relationship.

- **Material** indicates material transfer/exchange between two components. This interface can be a directional relationship.

- **Event** interfaces are required to verify that correct precedence relationships are obeyed throughout the assembly operation, or are used to help prevent a component from harming the operator or damaging itself by preventing one component from changing state due to the state of another component, and vice versa.

- **Electrical dynamics** interface represents any type of interactions between logical and physical components.

Please note that, the number and definitions of the interface types are dependent upon the context of the given design problem, and can be further broken down to achieve a finer level of abstraction.

6.4 Modelling static system complexity

Increasing number of components and numerous interactions with different kinds of flows exchanged between system components increases overall system complexity considerably. An increased complexity may result in reducing system safety, therefore, should be reduced without compromising the functional requirements. In this section, the Sinha [2014]'s analogy is revisited to evaluate static complexity of component-based assembly automation systems in a quantitative and repeatable manner. In here, it is hypothesized that the complexity of an assembly system is strongly linked to its inherent structural properties, and any improvement on the complexity has direct implications on the system's development and management effort.

Let's consider a system (S) composed of m number of physical and n number of logical components.In here, the inter-domain connectivity is defined by a domain-mapping matrix $K(mxn)$. Based on Sinha [2014]'s formulation of system-of-system expansion of the Huckel's theorem, the connectivity matrix of the resultant topology is defined as follows:

$$\Lambda = \begin{vmatrix} P & K \\ K^T & L \end{vmatrix} = \begin{vmatrix} P & 0 \\ 0 & L \end{vmatrix} + \begin{vmatrix} 0 & K \\ K^T & 0 \end{vmatrix} \tag{6.1}$$

where, Λ is the adjacency matrix of the resultant system-of-systems, P is the adjacency matrix of the physical system, L is the adjacency matrix of the logical system, K is the inter-domain connectivity matrix, and K^T is transpose of the inter-domain connectivity matrix. Accordingly, the graph energy of the overall system can be written as follows:

$$E_\Lambda = E_P + E_L + \Delta \tag{6.2}$$

where, E_P is graph energy of the physical system connectivity, E_L is graph energy of the logical system connectivity, and the term Δ represents the graph energy originated from the inter-domain connectivity. The resultant complexity of the system architecture C^S, thus, can be defined as follows.

$$C^S = C^S_P + C^S_L + C^S_\Delta \tag{6.3}$$

where, C_P^S is the complexity of physical system in isolation, C_L^S is the complexity of logical system in isolation, and C_Δ^S represents complexity induced by the inter-domain connectivity. By re-writing the original metric, the individual terms of the static complexity can be written as follows.

$$C_P^S = \sum_{i=1}^{m} \alpha_i^P + [\sum_{i=1}^{m}\sum_{j=1}^{m} \beta_{ij}^P] \frac{E_P}{m+n} \tag{6.4}$$

where, α_i^P is the complexity of the i^{th} physical component, β_{ij}^P is the complexity of the interface between physical components i and j, m is the total number of physical components, and n is the total number of logical components.

$$C_L^S = \sum_{i=1}^{n} \alpha_i^L + [\sum_{i=1}^{n}\sum_{j=1}^{n} \beta_{ij}^L] \frac{E_L}{m+n} \tag{6.5}$$

where, α_i^L is the complexity of the i^{th} logical component, β_{ij}^L is the complexity of the interface between logical components i and j.

$$\begin{aligned} C_\Delta^S = {} & [\sum_{i=1}^{m}\sum_{j=1}^{m} \beta_{ij}^P][\frac{E_L}{m+n}] + [\sum_{i=1}^{n}\sum_{j=1}^{n} \beta_{ij}^L][\frac{E_P}{m+n}] + \\ & [\sum_{i=1}^{m}\sum_{j=1}^{n} \beta_{ij}^K + \sum_{i=1}^{n}\sum_{j=1}^{m} \beta_{ij}^{K^T}][\frac{E_P+E_L}{m+n}] + \\ & [\sum_{i=1}^{m}\sum_{j=1}^{m} \beta_{ij}^P + \sum_{i=1}^{n}\sum_{j=1}^{n} \beta_{ij}^L][\frac{\Delta}{m+n}] + \\ & [\sum_{i=1}^{m}\sum_{j=1}^{n} \beta_{ij}^K + \sum_{i=1}^{n}\sum_{j=1}^{m} \beta_{ij}^{K^T}][\frac{\Delta}{m+n}] \end{aligned} \tag{6.6}$$

where, C_P^S and C_L^S are the complexity of physical and logical architectures, respectively. In the equation, static complexity of the resulted architecture is also growth by an additional element called as '*integrative complexity*' C_Δ^S, which is a direct result of two factors, i.e. complexity of inter-domain interfaces and graph energy of the inter-domain connectivity Δ [Sinha, 2014]. Please note that, the above mentioned elements of complexity are different in nature, therefore, overall system complexity should be always considered as a multi-dimensional value, rather than a single number. To achive this, Eq. 5.3 is corrected as follows:

$$C^S = w^P C_P^S + w^L C_L^S + w^\Delta C_\Delta^S \tag{6.7}$$

where, w^P, w^L and w^Δ represent weights of each element on the overall system complexity.

6.4.1 Component complexity, C_1^S

Component complexity C_1 represents the sum of complexities of individual system components, which are designated by α_i:

$$C_1 = \sum_{i=1}^{N} \alpha_i \tag{6.8}$$

Complexity of a component α_i represents the technical difficulty associated with the development activities of the component alone, not accounting the complexity of component's interfaces and the system's architectural information. This definition recognises the fact that the inherent complexity of a component is determined to a large extent by the people, hence can be termed as *subjective* complexity. This means that the same set of components can be judged differently by different designers, engineers and operators under different circumstances. This makes the issue of quantifying component complexity in an unambiguous manner, the so-called objective complexity, a real challenge. The important aspect is here to use a consistent estimation while comparing different structures in an objective manner through some of component attributes, which are easily and unambiguously observable and quantifiable.

6.4.1.1 Component complexity in the physical domain, α^P

Physical components are pieces of equipment that are bought from companies specialising in their design and manufacture, and are highly standardised both in their function and in their interfaces through which they interact with their surroundings [Lohse, 2006]. In this reserach, by following the approach proposed by [ElMaraghy and Urbanic, 2004], the underlying complexity of system components is associated with the information required to define/replicate the component. In other words, information is used as a representative for the relative effort required to use, operate, programme, control or interact with the component. It is assumed, in here, that physical system components are standard off-the-shelf products, which are ready for the system integration, as well as their inherent structures are hidden, and composed of a number of indecomposable parts. In light of this assumption, physical component complexity 'α_i' is defined in the form of an exponential function as follows:

$$\alpha_i = 1 - e^{-k(\sum_{j=1}^{N_i} c_j^{P,i})} \tag{6.9}$$

where, N_i is number of parts forming the component j, c_j^i is the information content of j^{th} part, and k is the exponential function parameter ($k \in [0,1]$), In here, an exponential function has been adopted in defining component complexity score as a result of two distinct reasons. The former is to scale complexity score between 0 and 1, thereby enabling a global range for all components. The latter is due to the fact that perceived complexity of an individual cannot be increased after reaching his/her limits of understanding. Thus, a negative exponential function can be used to shrink complexity score down to one, especially for components exceeding the limits of understanding. **Figure 6.3**, given below, is a surface map defining the relationship between total information content of the component and exponential function parameter k. This figure depicts that, for a constant k value, component complexity will slowly grow to positive 1, as its total information content increases.

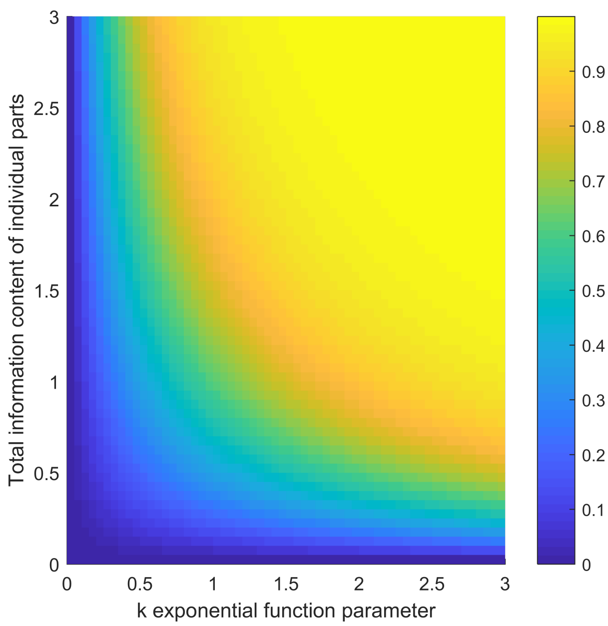

Figure 6.3: Surface plot of component complexity

In the model, $c_j^{P,i}$ is characterised through an indices-based methodology. A similar approach can be found in [ElMaraghy and Urbanic, 2004; ElMaraghy et al., 2010; Samy and Elmaraghy, 2011]. To this end, a set of attributes, that are typically associated to the technical complexity of a specific part, is defined by carrying out a series of workshops in collaboration with a group of machine builders (**Table 6.1**). Then, a Likert scale dividing each attribute over **four** categorical levels, is set up in

Table 6.1: Part attributes complexity factors

k	Attribute	#	Description	f_k^P
1	Size	1	Part is attainable and handy, and can be carried by hand	0
		2	Part has limited attainment or handling, or can be carried by two hands	1
		3	Part size is considerably large requiring more than one person or grasping aid	3
		4	Part is either extremely large or small, requiring special systems for manipulation and transport	9
2	Performance	1	Part has no performance tolerance	0
		2	Part has ordinary performance tolerances according to its class	1
		3	Part has tight performance tolerances and preponderance of only one function	3
		4	Part has tight performance tolerances of several specific functions	9
3	Maintenance	1	Part is maintenance free	0
		2	Part provides reproducible performance during its life-cycle, very little maintenance is required	1
		3	Part has a good reliability, but usually many constituents are involved in its maintenance	3
		4	Part is contamination sensitive, and/or requires regular (or frequent) maintenance	9
4	Functionality	1	Part has no structural/functional dependency	0
		2	Part satisfies functions of only one discipline i.e. structural, electric, thermal, etc.	1
		3	Part functions involve two coupled disciplines: i.e. electro-mechanical, etc.	3
		4	Part functions involve more than two coupled disciplines	9
5	Control	1	Part has no control feature	0
		2	Part is adjusted/operated by a human operator	1
		3	Part is automatically driven by an open loop control system	3
		4	Part is automatically driven by a close loop control system (both discrete and continuous)	9

order to have more control over the scaling of each attribute. In here, '*Quality Function Deployment*' (QFD) based approch is employed to indicate strong, medium, weak and non-existing interactions between the corresponding attribute and the relative information content of the equipment. It is adequate to use strong, medium and weak interactions as the initial step of the complexity estimation model in order to demonstrate the approach, especially in cases where the model is depended on the subjective decisions. The author believes that the suggested approach offers a reasonable beginning for future work, where more data and cases may be used to develop a better scale. In here, a four-point scale (9, 3, 1, 0) is used to represent the degree of interactions. In **Table 6.1**, '9' refers to strong interactions, '3' refers to medium interactions, '1' refers weak interactions and '0' refers no interactions. Moreover, it is assumed that the contribution of individual attributes on information

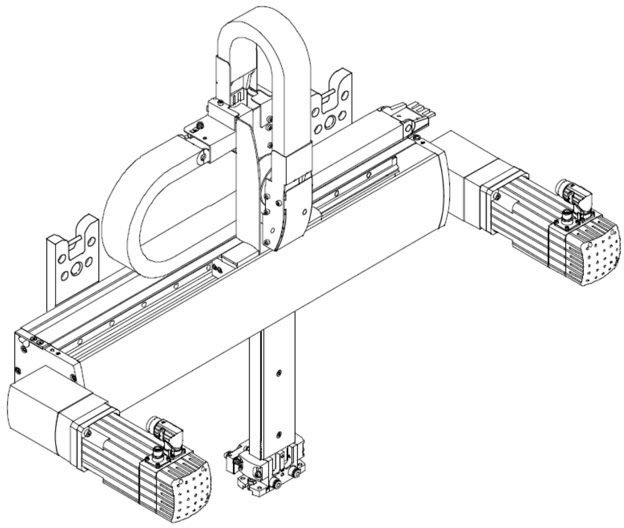

Figure 6.4: Linear gantry (Source: [AG, 2017])[Product information and schematic are taken from the original source].

content of a part is equally important. Accordingly, $c_j^{P,i}$ is calculated as below:

$$c_j^{P,i} = \frac{\sum_{k=1}^{n_a}(f_k^P)}{9n_a} \tag{6.10}$$

where, n_a is the total number of applicable attributes, and f_k^P is the corresponding value of k^{th} part attribute.

An example complexity assessment of a linear gantry illustrated in **Figure 6.4**, is given below. The gantry is used for fast repositioning of parts or modules in a large, rectangular working space in sorting, loading and unloading operations and is assumed to be a single physical component. Two fixed servo motors drive the toothed belts moving the slide of Y-axis and the interface on the Z-axis in a 2-dimensional space. The gantry also has a pneumatic parallel gripper which is positioned by a rotary electric drive. Complexity calculation of the linear gantry physical component is given in **Table 6.2**.

The gantry component described above, can also be defined using a more modular form, by defining a group of encompassed equipment of the gantry as a separate component if it has its own logical behaviour (e.g. modular configuration) or by representing them as separate physical component that are part of the gantry component if the component's parts are commissioned separately during manufacturing system built up phase (e.g. flexible configuration indicating that some parts of

Table 6.2: Complexity calculation of the linear gantry $k^P = 0.5$.

Equipment type	$N_j^{P,i}$	f_1^P	f_2^P	f_3^P	f_4^P	f_5^P	$c_j^{P,i}$	$\sum_{j=1}^{N_j^{P,i}} c_j^{P,i}$
Servo Drive (Electric)	2	0	3	1	3	9	0.356	0.712
Toothed Belt	2	0	0	0	1	0	0.022	0.044
Sensor (Proximity)	2	0	1	1	1	0	0.067	0.134
Rotary Drive (Electric)	1	0	1	1	3	3	0.178	0.178
Braking Resistor	1	0	0	1	1	0	0.044	0.044
Signal Converter	1	0	0	1	1	0	0.044	0.044
Sensor (Optic)	1	0	1	1	1	0	0.067	0.067
Drive (Pneumatic)	1	0	1	3	3	3	0.222	0.222
Work holder (Gripper Jaws)	2	0	0	0	1	0	0.022	0.044
Overall								0.363

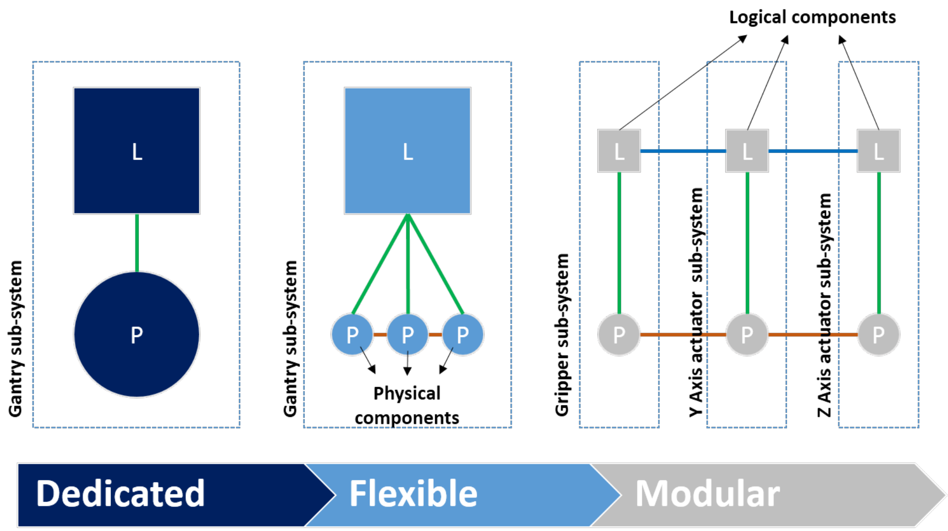

Figure 6.5: System components can be defined in various forms depending on the required level of modularity.

the component can be changed depending on the application requirements) (**Figure 6.5**). Modularity often results in complex system architectures as the system-wide information variety increases, but still may be desirable if system decomposability is of primary importance [Sinha, 2014].

6.4.1.2 Component complexity in the logical domain, α^L

Inherent structure of logical (control) components can be defined in various ways. In this research, by following the approach proposed by [Lee et al., 2007], these components are represented by a finite state machine (FSM). This approach is selected because of the data availability. According to Lee et al. [2007], "*a FSM is an abstract machine that can be in exactly one of a finite number of states at any given time*". FSMs can change from one state to another in response to external inputs; in here, the change between states is called a transition. A FSM is often defined by a set of states, including an initial state, and a set of conditions for each transition [Ahmad, 2014]. A FSM is composed of various control structures which are used to model the mental decisions, and alternative routing in the components sequence of operation. The simplest structure of a logical component is an execution control flow model, where the states have to be processed in a serial sequence. However, most FSMs includes split nodes and join nodes as well.

In a similar way, complexity of logical components is defined as the relative effort required to develop, maintain, and comphrened the FSM in a software engineering perspective, and estimated as follows:

$$\alpha_i = 1 - e^{-k^L(\sum_{j=1}^{n_i^L} c_j^{L,i})} \tag{6.11}$$

where, n_i^L is number of control structures exist in in i^{th} logical component, $c_j^{L,i}$ represents the relative cognitive effort required to develop j^{th} control structure, and k^L is the exponential function parameter ($k^L \in [0,1]$). In here, $c_j^{L,i}$ represents the relative cognitive effort required for comprehending (modifying or debugging) the corresponding control structure in the control flow diagram. These structures are described and illustrated in **Table 6.3**, where the equivalent normalised cognitive weight for each individual structure is defined based on the empirical studies in the cognitive and industrial informatics (please see [Shao and Wang, 2003]).

Table 6.3: Cognitive weights of FSM control structures.

Class	Control structure	$c_j^{L,i}$
States	Static/initial state	0
	Dynamic state	0.1
	Composite dynamic state	0.2
Transitions	Sequence transition	0.025
	XOR-split transition (exactly one of two branches)	0.1
	XOR-split transition (exactly one of ≥ 3 branches is chosen)	0.15
	OR-split transition	0.4
	Internal conditions	0.05
	Cancellation transition (by activating a state one deactivates another one)	0.2

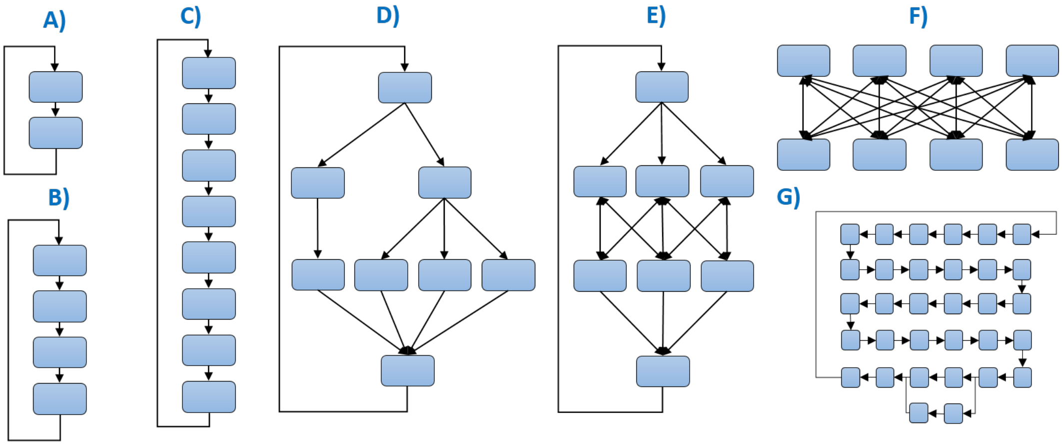

Figure 6.6: Example FSMs with varying degree of complexity.

Table 6.4: Example calculation of logical component complexities $k^L = 0.5$.

Design	States	Sequence	XOR-split (2 branches)	XOR-split (≥ 3 branches)	α^L
A	2	2	0	0	0.059
B	4	4	0	0	0.072
C	8	8	0	0	0.109
D	8	6	5	6	0.481
E	8	4	8	9	0.726
F	8	0	0	32	1.000
G	32	31	2	0	0.703

The complexity assessment approach is demonstrated using a number of FSMs with varying degree of complexity (**Figure 6.6**). In the example, states and transitions are represented by rectangles and directed arrows, respectively. To simplify the calculations, it is assumed that all states are dynamic, and transitions are either a sequence transition or XOR-split branch. **Table 6.4** shows complexity results of the given FSMs.

The results are found in accordance with the proposed hypothesis indicating that complexity increases as the FSM has more control structures. The author believes that the proposed measure can capture the intuition that a large but a well-structured FSM can be less complex, than a small but poorly structured FSM. Please note that, FSM complexity can be carried out in different methods. These include but not limited to: canonical measure (number of states), cyclomatic number (number of linearly independent loop) [McCabe, 1976a], and cognitive weight measure

[Shao and Wang, 2003].

Please note that, inherent complexity of system elements is defined as a relative effort required to develop/manage the component in an isolated condition, hence is a content-dependent property. In the content of assembly products (chapter 5), component complexity depicts the effort required to handle the component itself (which is induced by the component geometry and shape), whereas, in the content of assembly automation systems (chapter 6), component complexity, is a more broader phonemenon, defining ergonomical/physical efforts occur in various life cycle phases, e.g. deployment, control, maintenance and operation.

6.4.2 Pair-wise connection complexity, C_2^S

In the literature, complexity of pair-wise component interfaces are characterised by two essential elements; *i*) complexity of the interfaced components and *ii*) the nature of the connectivity c_k:

$$\beta_{ij} = (\sum_{k=1}^{l} c_k) max(\alpha_i + \alpha_j) \tag{6.12}$$

where, c_k is the interface coefficient defining the relative difficulty in establishing the interface type k (i.e. the nature of the connectivity), and l is the number of interfaces between components i and j.

This representation describes interface complexity as a fraction of the connected component complexities, such that interface and component complexities are not dimensionally mismatched. It is also reasonable, as the interface complexities are expected to be much smaller than the component complexities in cyber-physical systems [Sinha, 2014]. According to Sinha [2014], c_k can be estimated either using statistical methods or expert opinions. In this research, due to the lack of engineering data which is required to build a high-fidelity statistical model, expert opinions are used to estimate interface coefficients.

6.4.3 Complexity of the system's topology, C_3^S

As explained in the previous chapter, architectural pattern of a system leads to the topological complexity associated with the interactions between components and depends on the nature of the system's connectivity [Kinsner, 2010]. In the Sinha's model, topological complexity is defined as the matrix or graph energy E (see Nikiforov [2007]), which is calculated by the sum of singular values of the adjacency matrix of the system under consideration.

$$C_3^S = \frac{E_S}{n_s} \tag{6.13}$$

and

$$E_S = \sum_{i=1}^{N} \sigma_i \tag{6.14}$$

where E_S graph energy of the system under consideration, n_s is the total number of components and N is the total number of singular values of the adjacency matrix of system S.

According to Sinha [2014] this metric outlines "*the nominal effective dimension entrenched within the connectivity pattern of the system*", which is defined by the visualised by the binary adjacency matrix. In a practical manner, this quantity indicates the *intricateness* of structural dependency among system components [Sinha et al., 2017].

According to Sinha [2014], values of graph energy can be used to categorise different architectural patterns (**Figure 6.7**). Accordingly, the energy regimes for a system (A) with n number of components can be defined as: *i*) *hypo energetic*, *ii*) *transitional* and *iii*) *hyper energetic*. The hyper energetic regime is considered by the graph energy which is greater than or equal to that of the fully connected system,

$$E(A) \geq 2(n-1) \tag{6.15}$$

The hypo energetic regime is defined as:

$$E(A) \leq n \tag{6.16}$$

The intermediate regime between these two where the energy is higher than that of the hypo energetic regime and smaller than the hyper energetic is labelled as translational regime. These energy regimes can be translated into common architectural

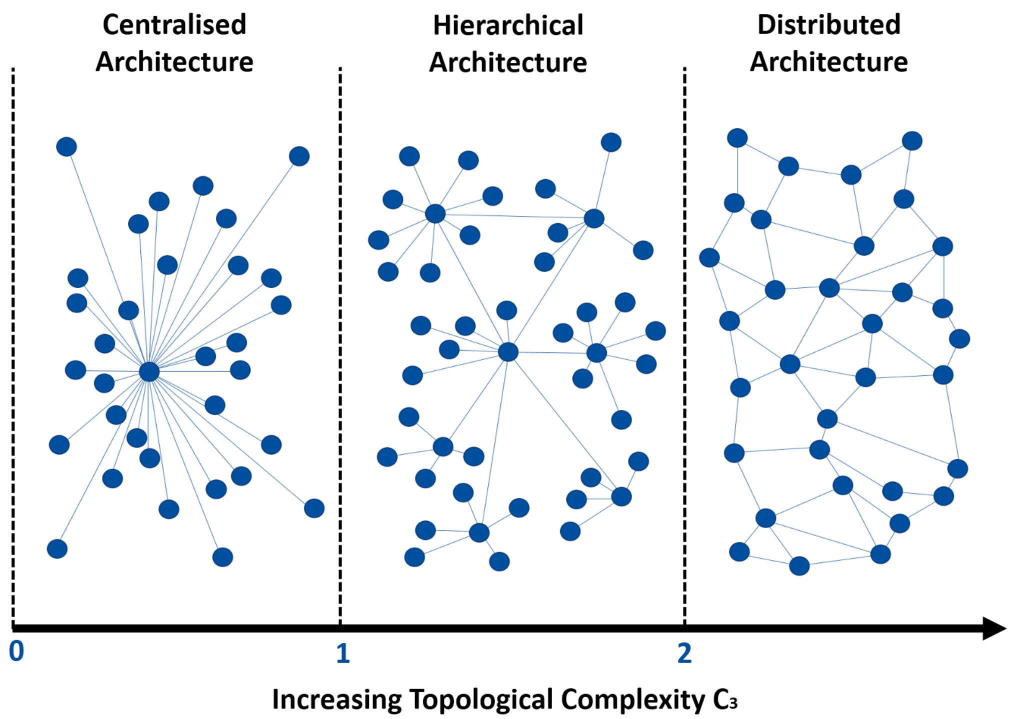

Figure 6.7: Spectrum of architectural patterns based on topological complexity (Source: [Sinha, 2014]).

pattern categories as follows:

$$C_3 = \begin{cases} \geq 2(1-\frac{1}{N}) \approx 2 & \approx \text{distributed architecture} \\ 2 > \ldots \geq 1 & \approx \text{hierarchical architecture} \\ < 1 & \approx \text{centralised architecture} \end{cases} \tag{6.17}$$

As it is understood from the expression given above, the topological complexity increases from centralised towards more distributed architectures.

6.5 Case study: Festo MPS

In this section, the complexity assessment approach is demonstrated using the Festo modular production system (MPS) (**Figure 6.8**). The Festo MPS is a laboratory based system mainly used for research and education purposes. This system reproduces the functionalities of assembly machines used at automotive assembly lines

[McLeod, 2013]. Main operation of the Festo MPS is to move work pieces from one end to another by performing a number of sequential operations, including as picking, indexing, drilling and storing [McLeod, 2013]. The test rig contains four subsystems, i.e. distribution, buffer, processing and handling, which are configured Schneider PLC and distributed I/O. The distribution unit consisting of a pneumatic feeder and a converter, forwards cylindrical work pieces from the stack to the buffer unit. Buffer unit consists of a conveyor system and a separator actuator, which are used to transport and separate out work pieces. After passing the buffer unit, work pieces are forwarded to the rotary table of the processing unit, where a drilling operation is performed. At the end, a handling unit removes parts from the processing station and sorts them according to their physical characteristics, e.g. shape and colour. A detailed description of Festo MPS can be found in [McLeod, 2013].

6.5.1 DSM formulation of the test rig

The first step in applying the theoretical framework is to develop the basic system DSM. In this example, system decomposition was carried out by following a methodology proposed by [Farid and McFarlane, 2007]. The inteface types were generalised into nine main categories, i.e. static structural, dynamic structural,

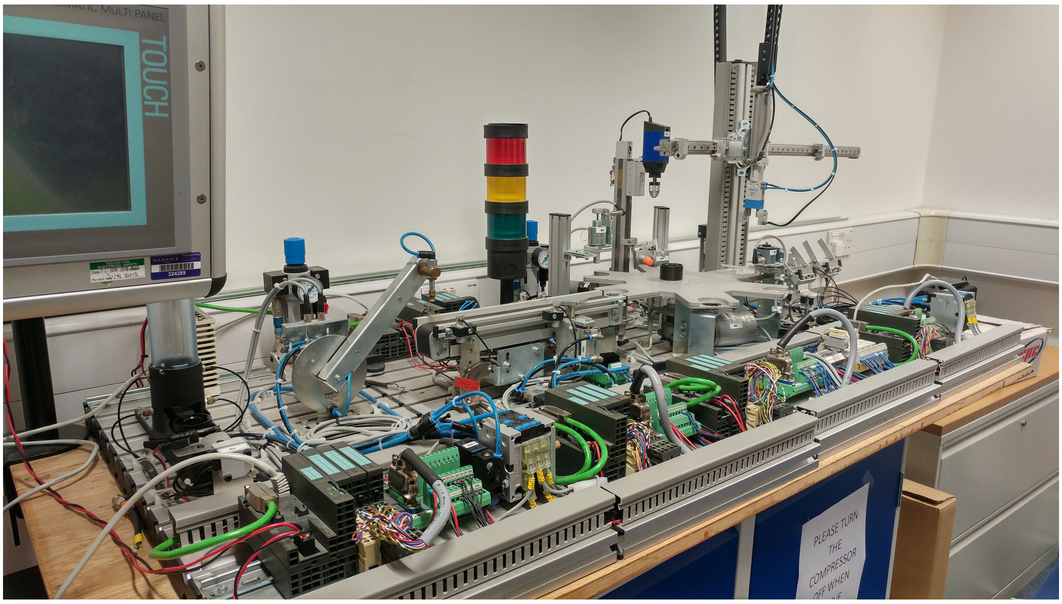

Figure 6.8: Festo MPS.

part transfer, fluid flow, energy transfer, spatial interactions, control/information exchange, and event flows (only within the logical architecture). Please note that, these interface types can be further broken down to increase the level of detail of the model. Furthermore, components such as electric and pneumatic energy supplies, start-up valves, and structural equipment, such as: cable holders, plate connectors, and electrical mounting systems, was not included within the DSM structure. The logical architecture of the Festo MPS was modelled as a network of logical components defining the abstract process behaviours of either one or more physical components. These components are designed based on the IEC 61499 standard ([Karl-Heinz John, 2003]), and deployed to a distributed PLC. The architecture contains process descriptions of 14 actuators, 13 sensors, and 7 process orchestrators, as shown in **Figure 6.9**. In here process orchestrator components are used to

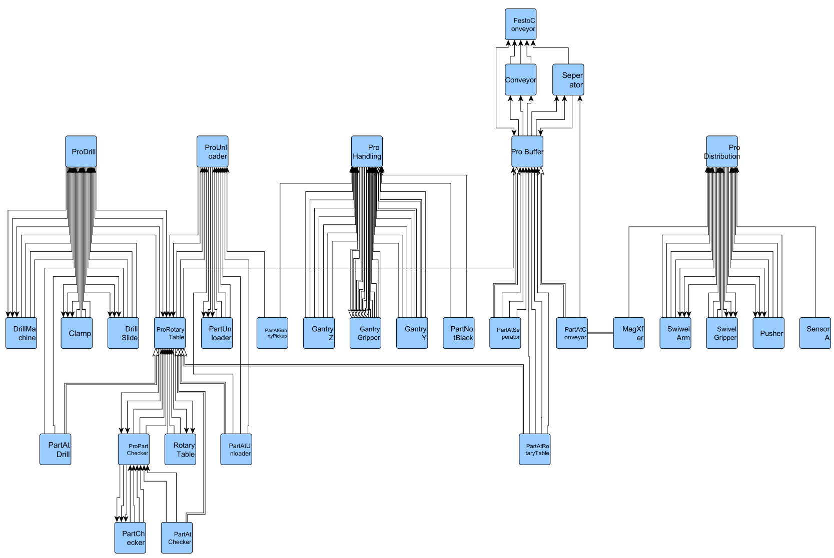

Figure 6.9: The logical architecture of the Festo MPS.

regulate a group of logical component. Please note that, physical and logical architecture of the test rig was mapped through electrical dynamics interfaces. **Figure 6.10** shows the clustered multi-domain matrix (MDM) of the Festo MPS test rig, where a marking in an off-diagonal cell indicates at least one interaction between two components.

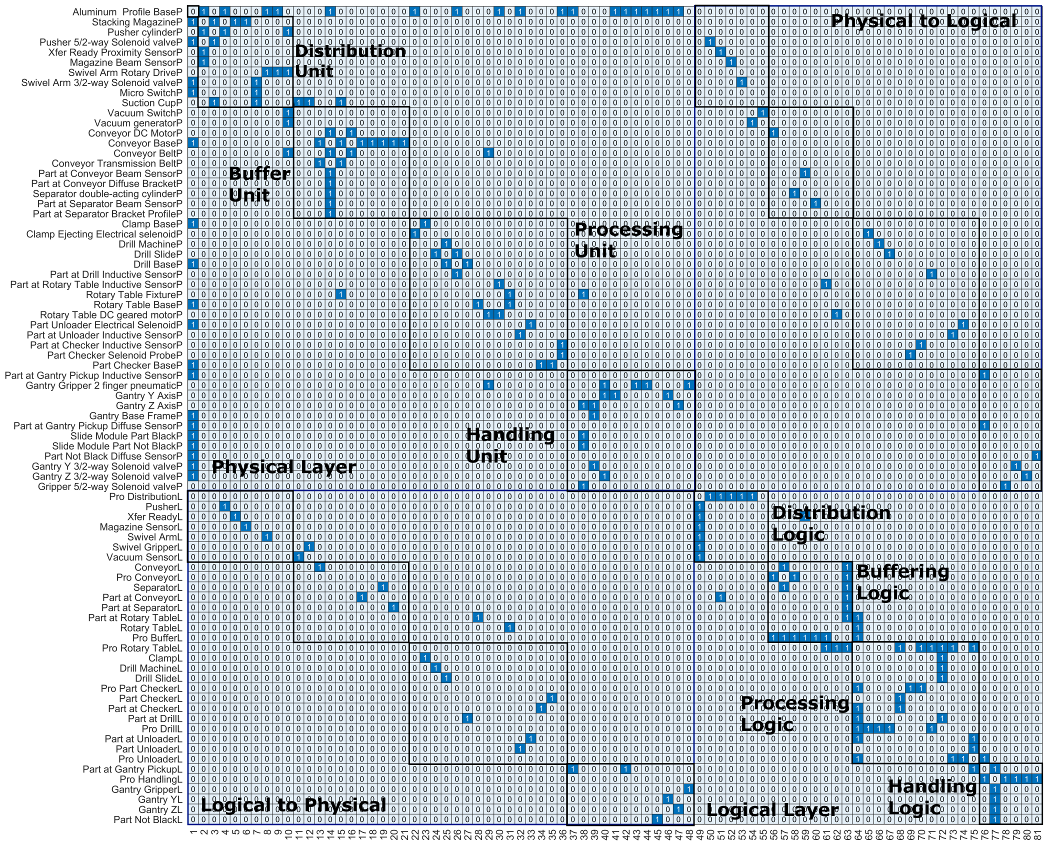

Figure 6.10: Multi-domain matrix representation of the Festo MPS.

6.5.2 Complexity estimation

Once the basic system DSM and corresponding component interfaces are defined, the proposed approach can be employed to assess static complexity of the overall system architecture, by following five consecutive steps detailed as below.

6.5.2.1 Step 1: Calculating topological complexity

Topological complexity indicates the *intricateness* of structural dependency among system components, and is calculated as the sum of singular values σ_i of the adjacency matrix of the system under consideration. Based on the DSM analysis, complexity C_3^S of the overall Festo MPS structure is found as 1.429 with a graph energy $E_{[MDM]^S}$ of 115.719 ($E_{[DSM]^P}$=53.004, $E_{[DSM]^L}$=33.490, and Δ=29.014). This indicates a hieararchical connectivity for the overall system architecture. The contribution of physical, logical, and interlayer topologies to the overall topological complexity is found to be 0.654, 0.413, 0.358, respectively. In a similar fashion, topological complexity of isolated physical C_3^P and logical system architecture C_3^L, without considering the effects of inter-domain connectivity, are found as: 1.104 and 1.015, respectively. This points out a transitional regime between hiearchical and centralised structure patterns for stand-alone physical and logical system architectures.

Now, let's consider the topological complexity of individual subsystems. In here, it is assumed that each subsystem is a stand-alone system that performs its functionality without having any inter-module interfaces. **Figure 6.11** compares the topological complexity of Festo MPS subsystems. According to the results, overall

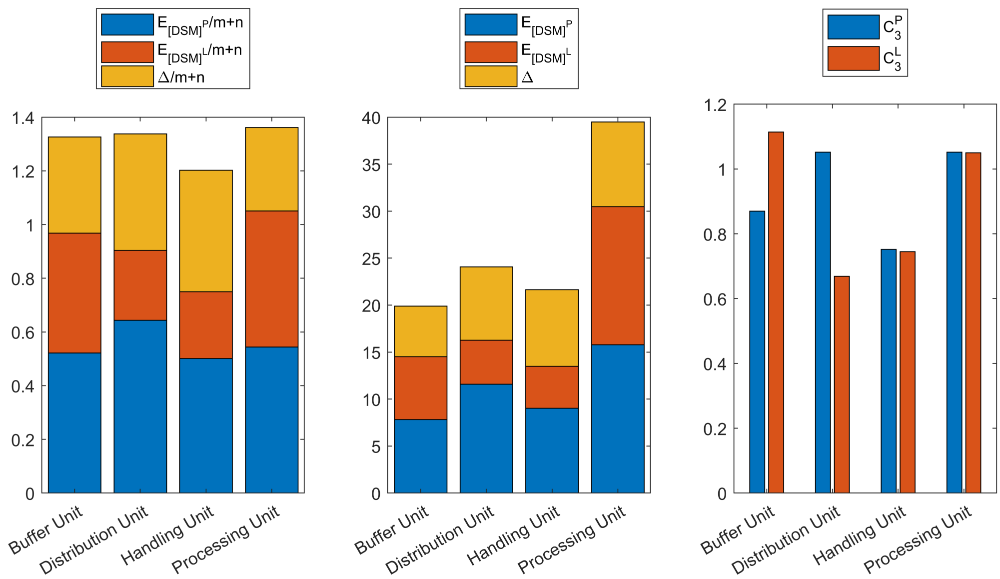

Figure 6.11: Comparison of Festo MPS subsystems: *left*: Overall topological complexity of subsystems, *middle*: graph energy results, *right*: topological complexity of isolated physical and logical architectures for each subsystem.

topological complexity for all modules was found to be above one. This indicates a hiearhical connectivity pattern for all cases. Interestingly, topological complexity of logical architectures (considered in isolation) for distribution and handling subsystems are found to be below 'one' indicating a centralised topology (**Figure 6.11** ***right***). This is reasonable as these subsystems are controlled by one process orchestrator (ProDistribution, and ProHandling), whereas, the number of logical components that are controlling the operation behaviours of buffer and processing subsystems are two (ProBuffer, and ProConveyor) and three (ProRotaryTable, ProDrill, and ProUnloader), respectively.

6.5.2.2 Step 2: Estimating physical component complexities

In the physical domain, structural composition of each component was documented through manufacturing bill of materials. In here, only actuators, sensors, valve regulators, fixture pins, work tools, electrical equipment, machine guardings, and magazines are considered as a component part. The relative information content of each equipment was calculated according to Eq. 5.10. Then, component complexities were determined according to Eq. 5.9. **Table 6.5** shows the total physical component complexities for Festo MPS and its subsystems.

Table 6.5: Total component complexities C_1^P in the physical domain.

	m	Estimate
Overall System	48	4.361
Distribution Subsystem	11	1.069
Buffer Subsystem	9	0.456
Processing Subsystem	15	1.288
Handling Subsystem	12	1.443

According to the results, the handling subsystem has higher total component complexity in the physical domain, whereas, the buffer subsystem has lowest complexity. This, in fact, is as expected since the handling subsystem comprises of components with larger information content, whereas the buffer system is quite simple. The method thus has potential to even differentiate the complexity of different systems with few components, thereby performing complexity assessment at a high resolution. **Table 6.6** shows the most likely estimates of physical component complexities.

6.5.2.3 Step 3: Estimating logical component complexities

In a similar fashion, complexity of logical components was estimated using the approach explained in **Section 5.4.1.2**. **Table 6.7** shows the results of total logical component complexities for Festo MPS and its subsystems. As reported by **Table 6.7**, processing subsystem has higher total component complexity in the logical domain. This is due to the fact that, processing subsystem has relatively more

Table 6.6: Complexities of individual physical components (Festo MPS).

Distribution Subsystem		Buffering Subsystem		Processing Subsystem		Handling Subsystem	
Name	α^P [0,1]	Name	α^P [0,1]	Name	α^P [0,1]	Name	α^P [0,1]
Stacking Magazine	0.044	Conveyor DC Motor	0.095	Clamp Base	0.022	Part at Gantry Pickup Inductive Sensor	0.064
Pusher cylinder	0.115	Conveyor Base	0.022	Clamp Ejecting Electrical selenoid	0.095	Gantry Gripper 2 finger pneumatic	0.259
5/2-way Solenoid valve	0.172	Conveyor Belt	0.026	Drill Machine	0.190	Gantry Y Axis	0.172
Xfer Ready Proximity Sensor	0.064	Conveyor Transmission Belt	0.026	Drill Slide	0.172	Gantry Z Axis	0.172
Magazine Beam Sensor	0.064	Part at Conveyor Beam Sensor	0.064	Drill Base	0.022	Gantry Base Frame	0.044
Swivel Arm Rotary Drive	0.172	Part at Conveyor Diffuse Bracket	0.022	Part at Drill Inductive Sensor	0.064	Part at Gantry Pickup Diffuse Sensor	0.064
3/2-way Solenoid valve	0.172	Separator double-acting cylinder	0.115	Part at Rotary Table Inductive Sensor	0.064	Slide Module Part Black	0.044
Micro Switch	0.064	Part at Separator Beam Sensor	0.064	Rotary Table Fixture	0.125	Slide Module Part Not Black	0.044
Suction Cup	0.064	Part at Separator Bracket Profile	0.022	Rotary Table Base	0.022	Part Not Black Diffuse Sensor	0.064
Vacuum Switch	0.043			Rotary Table DC geared motor	0.172	3/2-way Solenoid valve	0.172
Vacuum generator	0.095			Part Unloader Electrical Selenoid	0.095	3/2-way Solenoid valve	0.172
				Part at Unloader Inductive Sensor	0.064	5/2-way Solenoid valve	0.172
				Part at Checker Inductive Sensor	0.064		
				Part Checker Selenoid Probe	0.095		
				Part Checker Base	0.022		
Total	1.069		0.456		1.288		1.443

Table 6.7: Total component complexities C_1^L in the logical domain.

	n	Estimate
Overall System	33	5.2257
Distribution	7	0.9577
Buffer	6	0.8871
Processing	14	2.1666
Handling	6	1.2143

Table 6.8: Complexities of individual logical components (Festo MPS).

Distribution Subsystem		Buffering Subsystem		Processing Subsystem		Handling Subsystem	
Name	α^P [0,1]	Name	α^P [0,1]	Name	α^P [0,1]	Name	α^P [0,1]
Pro Distribution	0.393	Conveyor	0.139	Rotary Table	0.095	Part at Gantry Pickup	0.049
Pusher	0.139	Pro Conveyor	0.181	Part at Rotary Table	0.049	Pro Handling	0.588
Xfer Ready	0.049	Separator	0.139	Pro Rotary Table	0.221	Gantry Gripper	0.139
Magazine Sensor	0.049	Part at Conveyor	0.049	Clamp	0.139	Gantry Y	0.250
Swivel Arm	0.139	Part at Separator	0.049	Drill Machine	0.139	Gantry Z	0.139
Swivel Gripper	0.139	Pro Buffer	0.330	Drill Slide	0.139	Part Not Black	0.049
Vacuum Sensor	0.049			Pro Part Checker	0.286		
				Part Checker	0.139		
				Part at Checker	0.049		
				Part at Drill	0.049		
				Pro Drill	0.386		
				Part at Unloader	0.049		
				Part Unloader	0.139		
				ProUnloader	0.286		
Total	0.958		0.887		2.167		1.214

functionality (i.e. feeding, drilling, checking and unloading), and is composed of comparatively high number of logical components ($n = 14$). This is also in line with the prior hypothesis stating that functionality and complexity have a positive correlation. Accordingly, if a system has to perform a wide range of functionality or designed to support wide range of applications, it will likely have a complex structural composition. **Table 6.8** shows the most likely estimates of logical component complexities.

Table 6.9: Interface factors for Festo MPS.

Connection type	Interface factor, c_k
Mechnical structural	0.05
Mechanical dynamic	0.10
Spatial	0.07
Part transfer	0.10
Energy	0.10
Fluid flow	0.15
Control/information	0.20
Event	0.05

Table 6.10: Total interface complexity (Festo MPS).

	Estimate
Physical Interfaces	0.659
Logical Interfaces	2.532
Interdomain Interfaces	0.598

6.5.2.4 Step 4: Estimating interface complexities

In this research, two components are considered connected, if there is at least one connection exists between them. By considering all nine types of connections, an aggregated DSM of the Festo MPS was built. The aggregated DSM is a spanning subgraph of the basic DSM, which includes all connections. The representative interface factors for these connection types were determined by system engineers, and are listed in **Table 6.9** given below.

Complexity of pair-wise component interfaces was calculated by the approach presented in Section 5.3.2. Interface complexity results of Festo MPS are given in **Table 6.10**. It is important to note that, complexity of inter-domain interfaces was calculated by assumming the effects of the connected components on the interface complexity, are equally important.

In a similar fashion, interface complexity of each subsystem was estimated by neglecting inter-subsystem interfaces. **Table 6.11** shows the numerical results of total interface complexity for Festo MPS subsystems. It is noted from the table that, the handling subsystem has the highest interface complexity in the physical domain, whereas the buffer subsystem has the lowest interface complexity. It is interesting to note, although the processing subsystem has more number of physical components, its physical interface complexity is found to be lower than the handling subsystem's interface complexity. This is reasonable, as the handling subsystem contains a number of servo-pneumatic positioning systems requiring the integration of a series of fluid-flow connections, while, the processing subsystem is encompassing a number of stepper motors and electrical solenoids necessitating a series of electrical connections, which are relatively easier to develop and maintain. On the contrary, the processing subsystem has a relatively higher interface complexity on the logical domain. This is again, a result of the number of functions that the subsystem has to perform, i.e. more number of logical interlocks is required to control a wide range of applications in a synchronised manner.

6.5.2.5 Step 5: Estimating overall system complexity

As the final step, overall complexity of the system can be measured per Eq. 5.7. (w^P = w^L = w^Δ=1) **Table 6.12** shows the overall complexity of the Festo MPS test rig, respectively. According to the results, the system complexity is recorded as 13.205 (C^S_P = 4.792, C^S_L = 5.753, C^S_Δ = 2.660).

In a similar way, overall complexity of each subsystem is estimated as in **Table 6.13**. **Figure 6.12** displays the overall difference between categories with multiple complexity elements. As it is expected, the processing subsystem is found to be the most complex design (C^S = 4.40), whereas the buffer subsystem is found to be the simplest (C^S = 1.63). The complexity of the processing subsystem is a result of the logical architecture rather than the physical system as seen from the Figure. The approach, in addition to providing the overall system complexity value, is capable of indicating the source of complexity with a good degree of resolution. It is to be noted that, if the logical architecture has high value it is to be expected that the programming of the process sequence and its logic will be complicated. On the other hand, a high value of physical system complexity represents the difficulty in integrating the associated components. Furthermore, the results of the approach were presented to the engineers who were involved in the system build and based on

Table 6.11: Total interface complexity (Festo MPS Subsystems).

	Distr.	Buffer	Process	Handling
Physical Interfaces	0.105	0.024	0.065	0.1734
Logical Interfaces	0.391	0.154	0.854	0.675
Interdomain Interfaces	0.141	0.0714	0.200	0.138

Table 6.12: Overall system complexity (Festo MPS).

	Overall	Physical	Logical	Integrative
Estimate	13.205	4.792	5.753	2.660

Table 6.13: Overall subsystems complexity (Festo MPS).

	Overall	Physical	Logical	Integrative
Distribution subsystem	2.886	1.136	1.064	0.686
Buffer subsystem	1.673	0.468	0.942	0.263
Processing subsystem	4.980	1.324	2.600	1.057
Handling subsystem	3.844	1.530	1.382	0.932

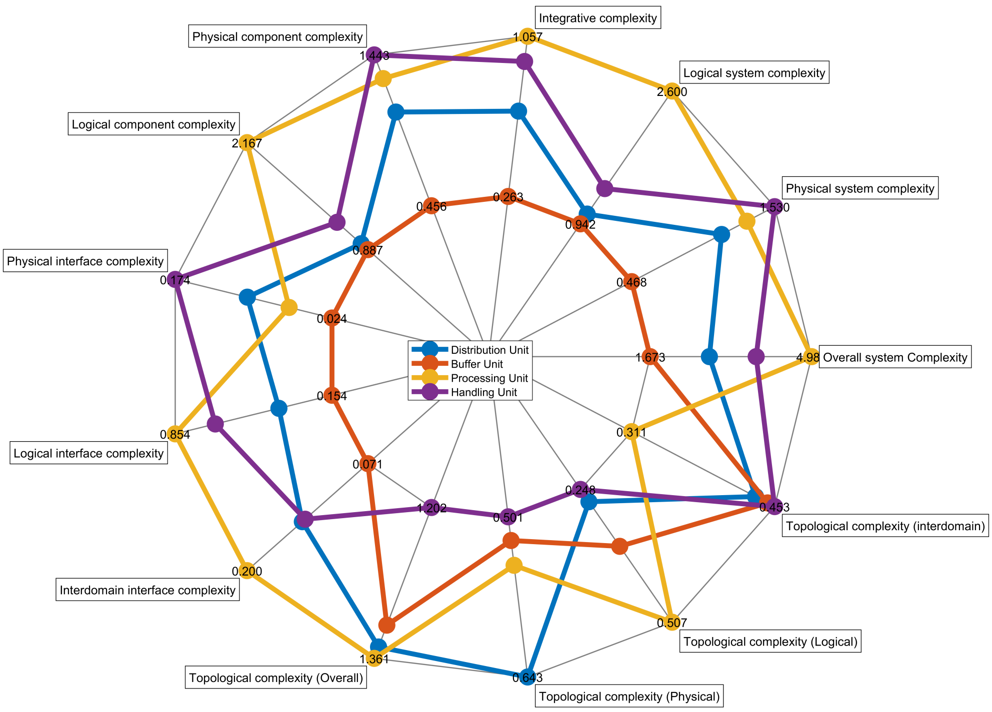

Figure 6.12: Complexity comparison of the Festo MPS subsystems

the feedback it is understood that, the presented complexity values are in agreement with the numbers that the engineers intuitively propose as the system complexity.

6.5.3 Discussion

In this section, Festo didactic test rig was used to demonstrate and provide a first-hand evaluation of the proposed complexity assessment method. The results showed that, the approach is mathematically rigorous, and assess static complexity at a high resolution over a broad spectrum ranging from topological complexity to physical and logical system components. Although the benefits of the model have been quantified with the help of the test case, some disadvantages that were identified are discussed below.

- Firstly, the approach requires relatively big amount of data, and hence may be **costly** to implement, especially in **large scale manufacturing systems**. Owing to the necessity of collecting immense quantity of data, in addition to the analysis, the approach can become demanding and laborious.
- Secondly, the proposed approach may be sensitive to the selected level of system decomposition. In other words, if two different systems are decomposed at different levels of granularity (i.e. coarse and fine decompositions), it would not be possible to compare them. Therefore, it is necessary to perform the comparison by establishing a standard during the modelling, to ensure comparison is done across similar level of granularity.

The above-mentioned issues can be addressed with the help of standardised engineering tools (e.g. virtual engineering), wherein the process of data collection can be automated, thereby eliminating the need to perform laborious tasks. Additionally, the use of standard for system decomposition can overcome the problem associated with the sensitivity to level of decomposition. In this regard, the chapter six will discuss the realisation of a proactive design support framework, wherein the use of a virtual engineering toolset to support the design rationalisation in the system architecting phase is detailed.

6.6 Chapter summary

In this chapter, a methodology for asssessing the complexity of assembly automation systems was presented. The approach was implemented using a modular production system and the capability of the methodology to identify the complexity of physical, logical and their integrated architecture was highlighted. Additionally, the limitations of the approach were identified and briefly explained. To overcome the disadvantages, the integration of the methodology with virtual engineering is suggested as a potential solution.

Chapter 7

Complexity-inclusive design support framework

As described in the previous chapter, the use of theoretical complexity models can be time consuming and tedious, especially in large scale design projects, where a significant amount of data collection and analysis are required. Hence, there is a need for practical tools and methods that designers and managers can use concurrently with the design process, so that, conceptual designs can be improved, or compared with various design alternatives for a better design solution. This chapter presents a complexity-inclusive design support framework which is achieved by the integration of theoretical model explained in the previous chapter, with a virtual system design and development software, namely: the vueOne virtual engineering (VE) tool. **Figure 7.1** shows an overview of the complexity-inclusive design support framework.

In the proposed framework, virtual design data generated at the vueOne system modelling phase are streamlined into a MATLAB application. This application uses a standard vueOne input, which can be directly extracted from the tool, and provides complexity results in textual and graphical formats. This results can be used together with the common design indicators, such as: cost, modularity, reconfigurability, energy consumption, etc., to achieve better decision-making at the conceptual design stage. In this chapter, vueOne VE tool-set, and the developed MATLAB application are described in detail. Moreover, the application is tested using three virtual assembly system designs modelled in the vueOne VE tool. The results showed that the proposed framework can help designers to improve/modify

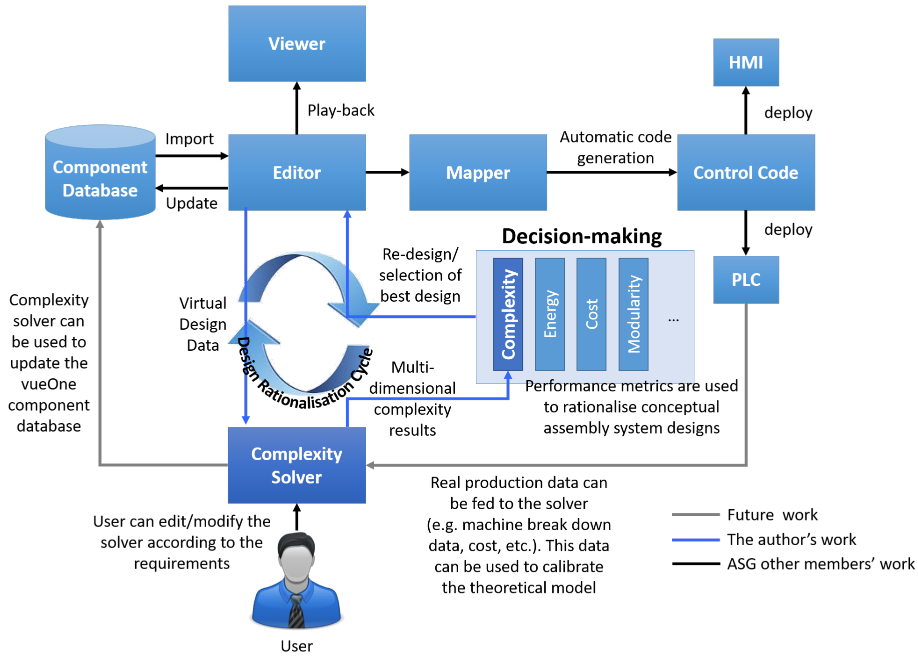

Figure 7.1: Complexity modelling and management framework integrated into the vueOne virtual engineering tool.

the system designs by comparing them with possible alternatives in a more practical way than the pen-and-paper based complexity assessment techniques. Finally, the proposed framework is statistically validated using a logistic regression model.

7.1 Virtual engineering

As one of the key enablers of agile manufacturing, virtual engineering (VE) is often used in the *system architecting* phase, where various system development activities such as; layout planning, assembly process planning, machine programming through auto-code generation, etc., can be performed with a computer model of the actual or planned setup, which may or may not physically exist. Moreover, Manesh and Schaefer [2010] states that "*the visualisation feature enabled by 3D simulations offers not only closer-to-reality information for users, but also enables new application domains to be addressed such as the rapid prototyping of machine systems*".

In todays manufacturing environment, VE activities can be realised with dedicated software, where workstation configurations can be evaluated/generated by integrating different components from a component library. As an example, Siemens PLM Software [Siemens, 2003] can be used in achieving full traceability of systems from their initial design to final manufacturing and/or optimise manual operations using digital human models, automatic validation of assembly/disassembly processes, simulate and validate the flexibility of the manufacturing lines before actual production. Another complete VE solution which can be used as an engineering tool in defining, generating and simulating the assembly and manufacturing process is Delmia [Del]. Delmia enables the possibility to the designers to perform activities such as: assembly/manufacturing feasibility studies; define the issues that appear; generate and optimise the manufacturing and assembly processes.

Although these software packages offer important capabilities, there is currently no concurrent complexity assessment functionality in such virtual system design and development environments. In fact, one of the major goals in system development, is to manage static complexity so as to keep the dynamic and emergent complexity of the system well understood and under control. Also, complexity is directly proportional to cost of the design and its implementation effort. Therefore, measuring and understanding of the complexity of a system architecture, concurrently with its design process, is very important for the whole system development enterprise. Similar practices can be frequently applied to the software design and development processes, where software architectures are simultaneously analysed using complexity metrics to keep project budgets under control.

7.2 vueOne virtual engineering tool

This research utilises the vueOne virtual engineering tool developed by the Automation Systems Group (ASG) in the University of Warwick to fulfil the current gap between the theoretical formulation of complexity metrics and their practical applicability to real-world system development. The vueOne tool is designed upon the "*component-based*" design paradigm, and is primarily used for the virtual commissioning of manufacturing systems supported by integrated components which are dedicated to performing a set of specific functions [Ahmad, 2014]. **Figure 7.2** shows the production system life-cycle achieved by using vueOne tool set. In the vueOne, a component is defined as a reusable, reconfigurable building blocks of the production system, providing a data integration mechanism for control be-

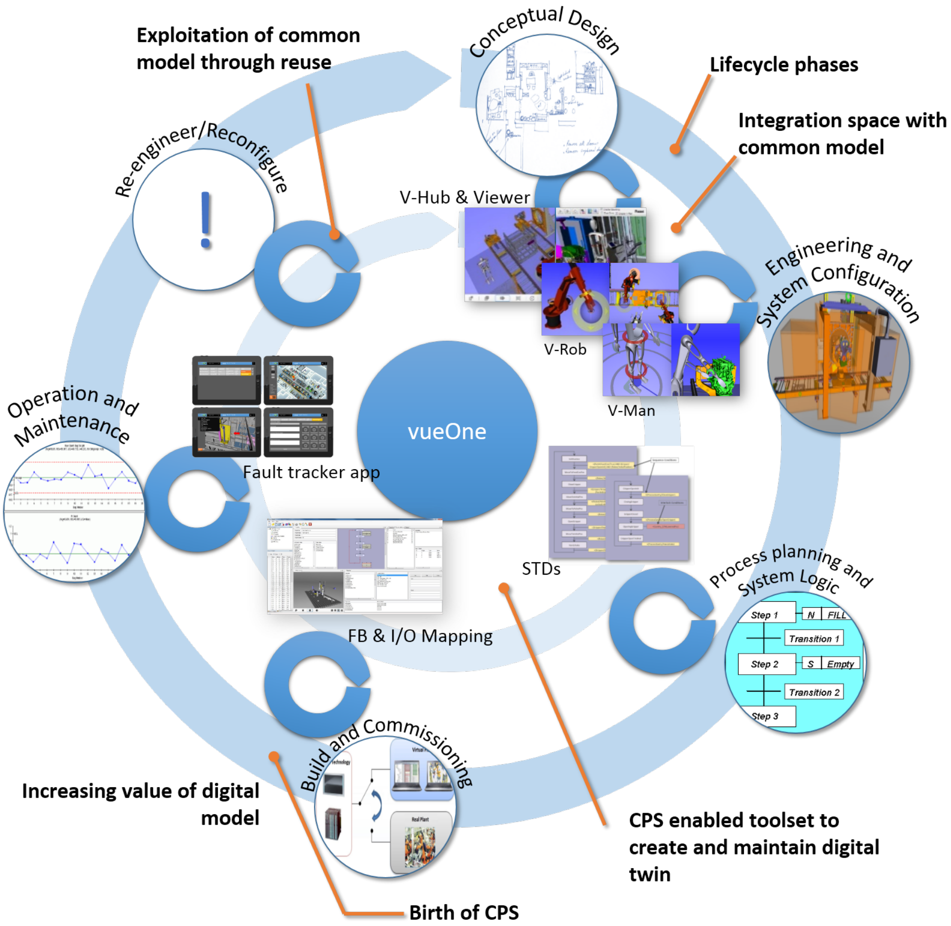

Figure 7.2: Manufacturing system life-cycle phases supported by the vueOne virtual manufacturing tool (Source: [Konstantinov et al., 2017]).

haviour, kinematics, geometries, and other data types defining a particular system resource [Ahmad et al., 2015]. Data that is encapsulated within a component can exist at a particular level of granularity which is defined by the user [Ahmad, 2014]. Presently, vueOne tool set delivers functions such as: 3D modelling, process simulation and evaluation, auto-control code generation, but not yet complexity assessment functionality.

The vueOne uses the standard Virtual Reality Modelling Language (VRML) format for 3D modelling, and common state-transition diagrams for component logic editing and visualisation. The vueOne tool set supports modelling of several types of components such as: sensors, actuators, digital human workers, robots and fixtures [Ahmad, 2014]. In the tool, a common component architecture is used to integrate component geometry, kinematic and control behaviours. Each component created by the tool has a unique ID which can be used in identification and debugging purposes. Components and systems can be stored in the library with any information associated with component parameters, and can be reused any number of times. The stored information can also be used to linked component performance. Modelling and simulation tasks in the vueOne tool, are performed within two steps: *i*) component modelling and *ii*) system modelling.

7.2.1 Component modelling

In the tool, components are modelled using the component builder module. This module enables supports the generation of component geometry, animation of its physical definition using the control behaviour state-transition diagram and kinematics [Ahmad et al., 2015]. There are six component categories, namely: actuators, sensors, virtual, non-control, virtual manikin (V-Man) and process logic. Actuator components accept data signals and perform the respective actions, sensor components obtain data from the external environment and change state accordingly, virtual components are used to represent abstract items for timing or algorithms [McLeod, 2013]. Non-control components often define machine frameworks or the work pieces involved. They do not have any inputs or outputs. A virtual manikin represents the human operator, and process component is used to define and regulate a group of components [Ahmad et al., 2015].

The vueOne tool enables a web-based collaborative engineering approach by allowing the imports of neutral CAD formats i.e. STEP to VRML [Ahmad et al., 2015]. The imported CAD geometry is stored in a library which expressively decreases the memory requirements. The imported geometries are then integrated to each other by means of link points. In here, the link points are the locations where multiple geometries joined together to form a complete component. Component behaviours can also be modelled by defining parameters, such as: kinematic relationships between geometries (translational or rotational), duration of the movement, average velocity, etc.

In the vueOne, component behaviours are defined through state transition diagrams (STDs) that are broadly compliant with IEC 61131-3 [Karl-Heinz John, 2003], and so PLC code can be automatically generated and deployed to support a basic level of virtual commissioning. The STD within the vueOne has three types of states: *i*) initial state, *ii*) static state, and *iii*) dynamic state. The component specific control parameters can also be employed in the vueOne tool. These parameters include velocity, torque, distance, etc, and can be changed to suite specific system requirements by adjusting them without affecting component's abstracted behaviour. The component modelling overview is shown in **Figure 7.3**.

7.2.2 System modelling

The system builder module is used to create a complete manufacturing system that includes component interfaces to define how each component will interact with other components in the system. Components are added to the system from the library, and assembled through the link points. The resultant configuration can be animated through the use of the *state viewer* to prove the correct assembly and operation of the manufacturing system. In the logical side, interlocks are added to the states of each component to defined how they interoperate with respect to the behaviour of other components within the system. The interlocking can be done

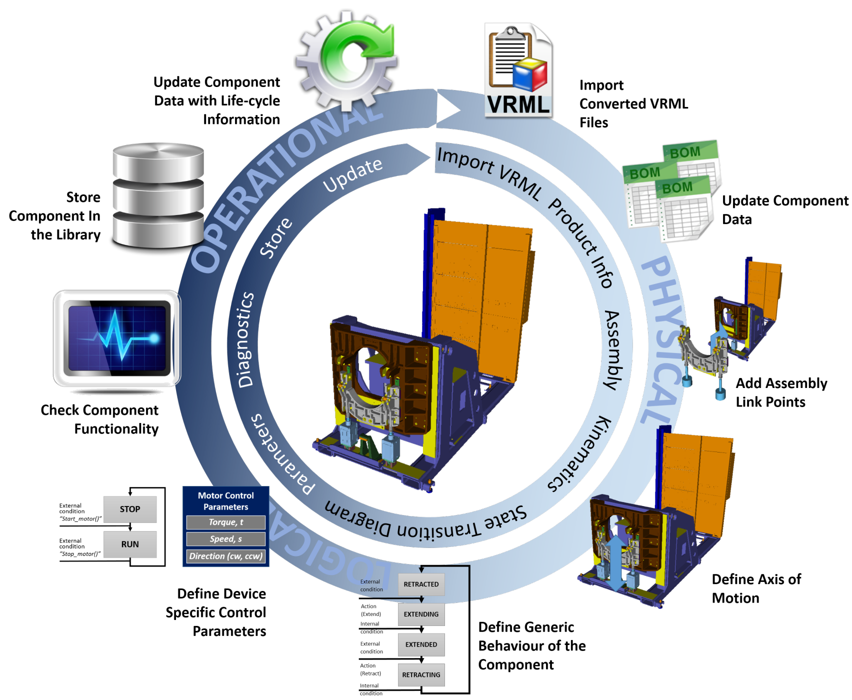

Figure 7.3: Component life-cycles in the vueOne.

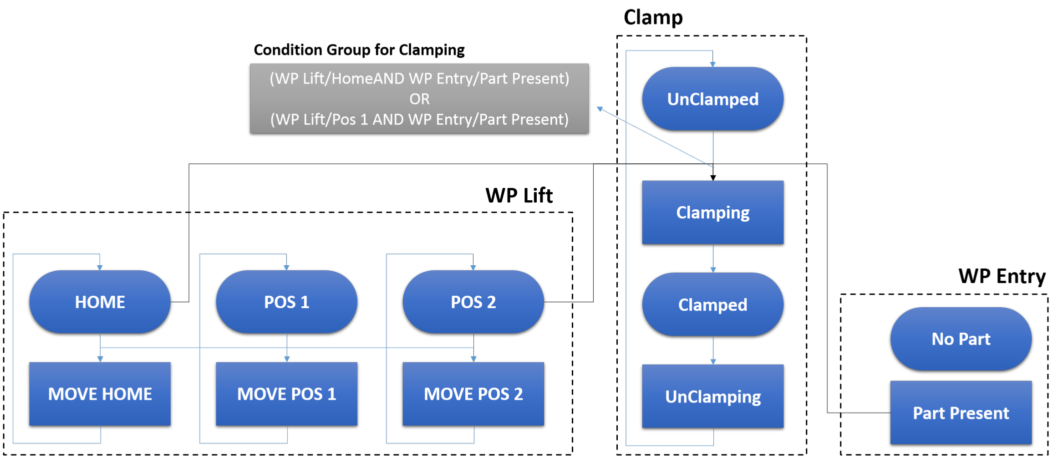

Figure 7.4: Component sequence interlocks (Source: [McLeod, 2013]).

through either sequence checks and safety interlocks. A sequence check is required to verify that correct precedence relationships are obeyed throughout the manufacturing operation. A safety interlock, on the other hand, is used to help prevent a physical component from harming the operator or damaging itself or other components by preventing one component from changing state due to the state of another component, and vice versa. Once the necessary conditions are satisfied, the state machine then defines the reaction of the component on input events in a given state according to the assigned operation parameters. An example logical architecture of a simple system is given in **Figure 7.4**. This system includes one sensor (i.e. WP Entry) and two actuators (i.e. WP Lift and Clamp), and the condition is defined to allow the clamp to progress from the 'Unclamped' to 'Clampin' states. Interlocks entered for the clamp to start 'clamping' is: '*(WP Lift/HomeAND WP Entry/Part Present) OR (WP Lift/Pos 1 AND WP Entry/Part Present)*'. Interlocks are entered as condition groups. Each condition within the group is an AND between each is an OR.

In order to simulate the behaviour of a manufacturing system, external inputs are also required. These inputs enable the effects of a part as it enters, move through and exits the system to be simulated and visualised. The vueOne uses the knowledge gained from the CAD models and engineers' experience to predict which sensors will be triggered, as the work piece travels through the system. In the vueOne, this is achieved by creating a 'Work-piece Routing Logic' that sets sensor values, moves work-pieces from link point to link point and sets the work-piece visibility. The work-piece routing logic may include decision box, in the routing chart, to direct flow based upon the work-piece. This redirection of the work-piece routing allow different work-piece variants to trigger different sensor states without the need for separate routes for each variant.

7.3 The complexity solver

The complexity solver is an add-on application to the vueOne toolset which is developed to prove the capabilities of the complexity-inclusive design support framework. It was developed by the author in the MATLAB programming language. The solver has three main modules: complexity engine, complexity database and graphical user interface. The structural modules of the applications and its interactions with the existing vueOne tool is shown in **Figure 7.5**. Currently, the solver requires both the vueOne toolset and user inputs, to estimate static complexity of virtual system designs, but, with further integration and enhancement, it is feasible to automatically extract the required information from the vueOne toolset directly. The required data sets by the complexity solver are listed in **Table 7.1**. Once the virtual design data and the end-user specifications are imported and defined, the application estimates the complexity of the virtual assembly system design according to the approach proposed in the previous chapter.

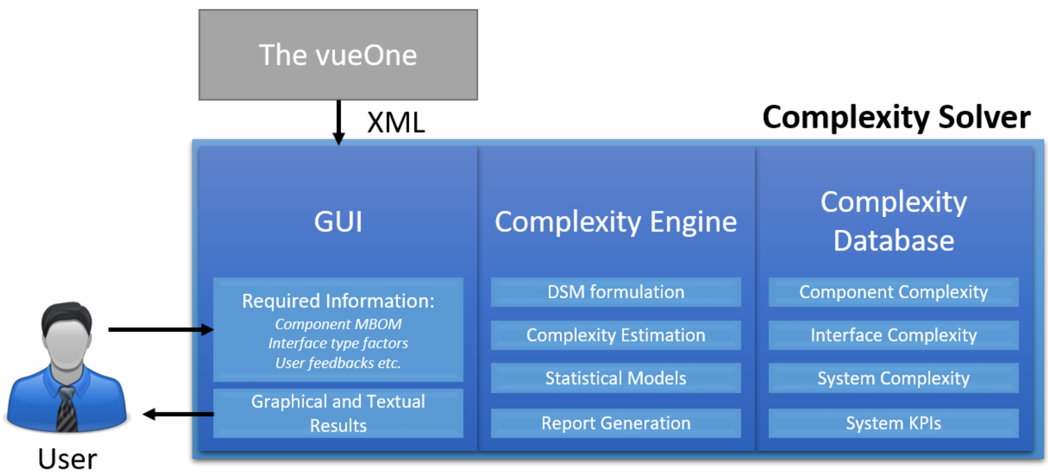

Figure 7.5: The modules of the complexity solver and its interactions with the existing vueOne tool and users

Table 7.1: Required inputs by the MATLAB application.

	Input	Description	Source
1	Component Tag	ID, name, manufacturer, year, etc.	vueOne XML
2	Component Class	Control oriented categorisation, i.e. controlled, non-controlled, process, etc.	vueOne XML
3	Component Type	Function oriented categorisation, e.g. motion, joining, holding, virtual, etc.	vueOne XML
4	Component Sub-type	A finer level of categorisation, e.g. measurement: optical sensor, vision system, etc.	vueOne XML
5	Component FSM	Sequence of operation of the component	vueOne XML
6	Logical interfaces	Events recevied and announced by the component (safety interlocks, etc.)	vueOne XML
7	Mechanical interfaces	Pair-wise mechanical interfaces between system components	vueOne XML
8	Interface type factors	Complexity calculation of pair-wise interfaces	User
9	Manufacturing bill of materials	Complexity calculation of physical components	User
10	System KPIs	Calibration of the model (not yet implemented)	User/Plant

Figure 7.6 shows the flow diagram of the complexity estimation process in the developed complexity engine. In this approach, the complexity engine first decomposes vueOne components into their physical and logical constituents. Accordingly, complexity of component FSMs is calculated based on the approach explained in **Section 6.4.1.2** and stored in the logical component complexity matrix ($[CCM]_L$). This process continues until complexity of all controlled components

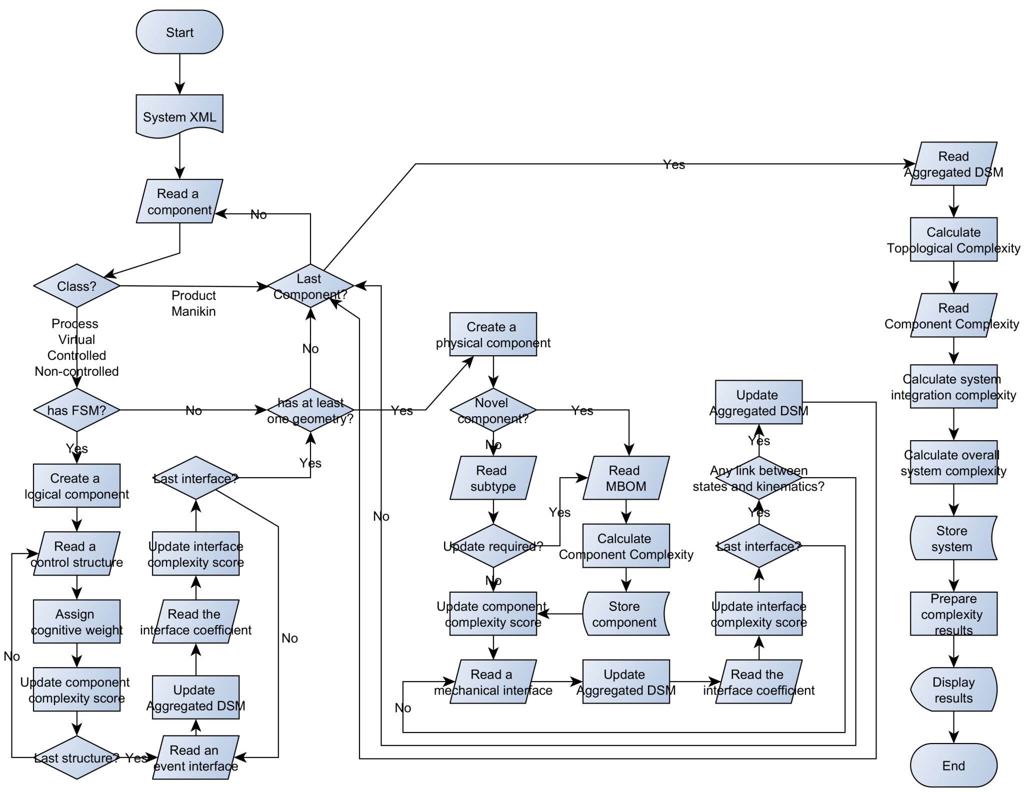

Figure 7.6: Workflow diagram of the complexity engine

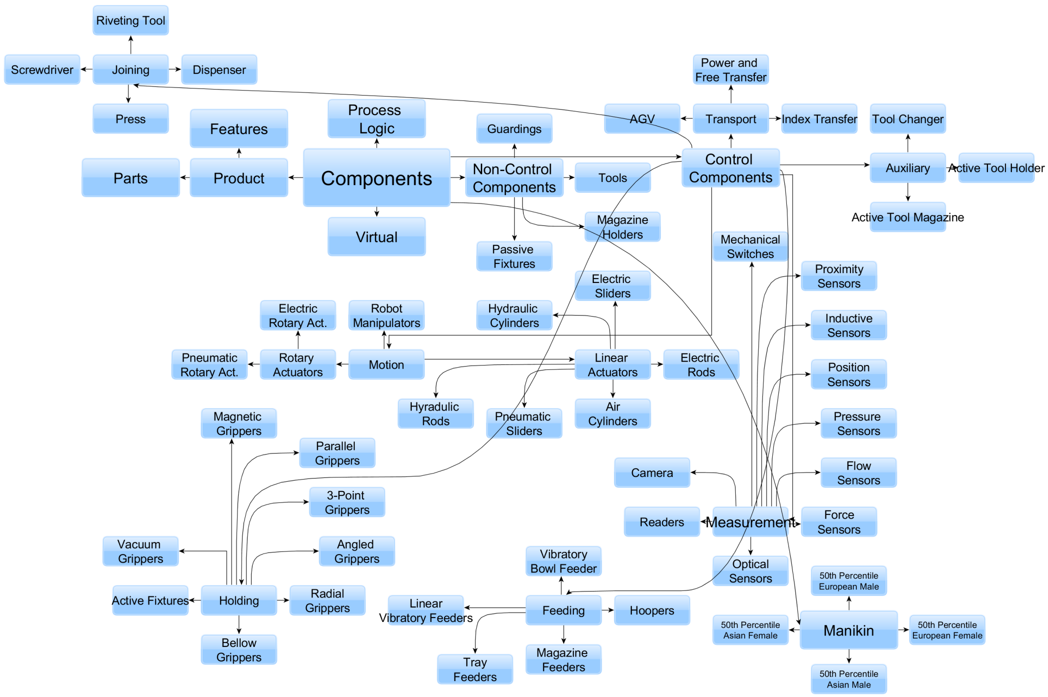

Figure 7.7: The vueOne component classes and types

are calculated and stored. Although the cognitive weights of each control structure is predefined within the engine, the designer can also manually edit these values through the GUI. In a similar way, physical compositions of vueOne components are analysed through the approach presented in the **Section 5.4.1.1**. For ease of calculation here, predefined complexity scores are assigned to each component subtype, e.g. $opticalsensor = 0.064$, $AGV = 0.589$, etc. Component subtypes used in the vueOne tool are depicted in **Figure 7.7**. However, in the case of bespoke components within any subtype, these predefined values can be changed by editing/correcting component's bill of materials through the developed GUI (**Figure 7.8**). These components can be stored in the solver database, under a specific component ID for reuse. Physical component complexities are then stored in the physical component complexity matrix ($[CCM]_P$). Please note that, vueOne XML output currently does not contain subtype information of system components. Therefore, this information should be manually defined by the end-user through the GUI, for each component modelled within the simulation.

Logical interfaces can be directly captured from the vueOne XML output. In this ouput file, each sequence check and safety interlock between two FSMs, is

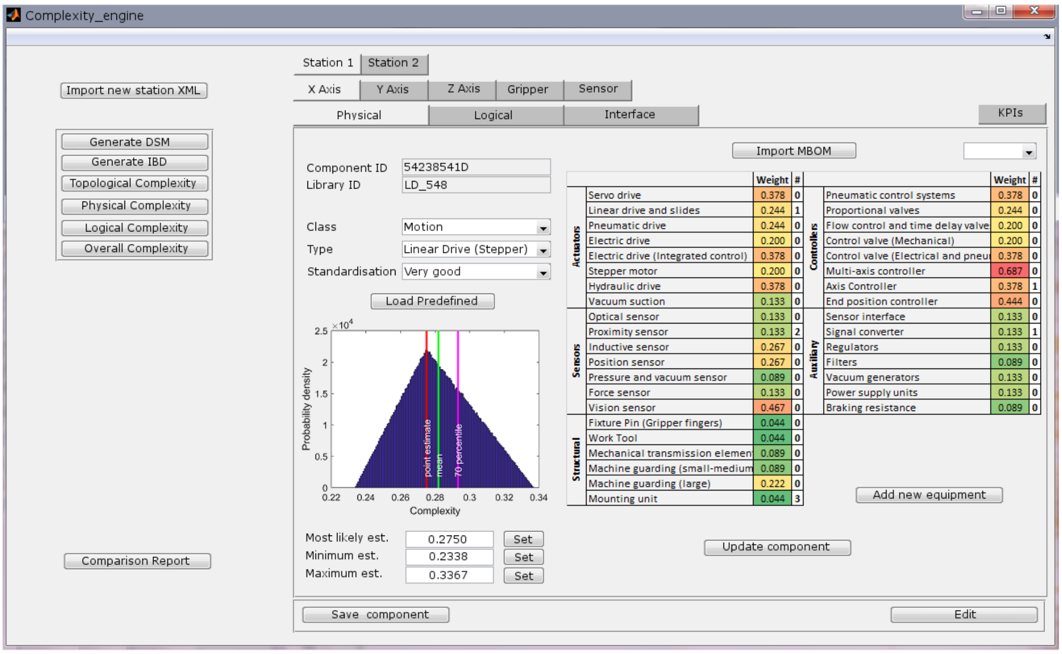

Figure 7.8: Complexity engine GUI.

considered as an event interface. Consequently, the logical connectivity pattern of the virtual assembly system design is stored in the logical interface connectivity matrix ($[ICM]_L$). This matrix contains information such as: connected component pairs, number of event interfaces and their types.

In the physical domain, the vueOne VM toolset only supports modelling of structural and kinematic relationships. These relationships include: static structural connections, dynamic rotational and dynamic translational kinematics, spatial interactions, and product/part exchanges. These interfaces are defined through link points, which are attached between component VRML geometries. In the current vueOne XML output, information regarding linkpoint types and connected geometries are not available, hence, they are directly extracted from the vueOne VRML editor, via a comma separated values (cvs) text file.

Please note that, above mentioned data collection issues, can be addressed through developing a new XML data structure. In here, an XML structure namely, *Complexity*.XML is proposed as a potential solution. This new data structure contains all required information that can be directly extracted from the vueOne toolset, and can be added to the tool once the proposed framework is fully integrated into the existing vueOne tool. *Complexity*.XML data structure is shown in **Figure 7.9**.

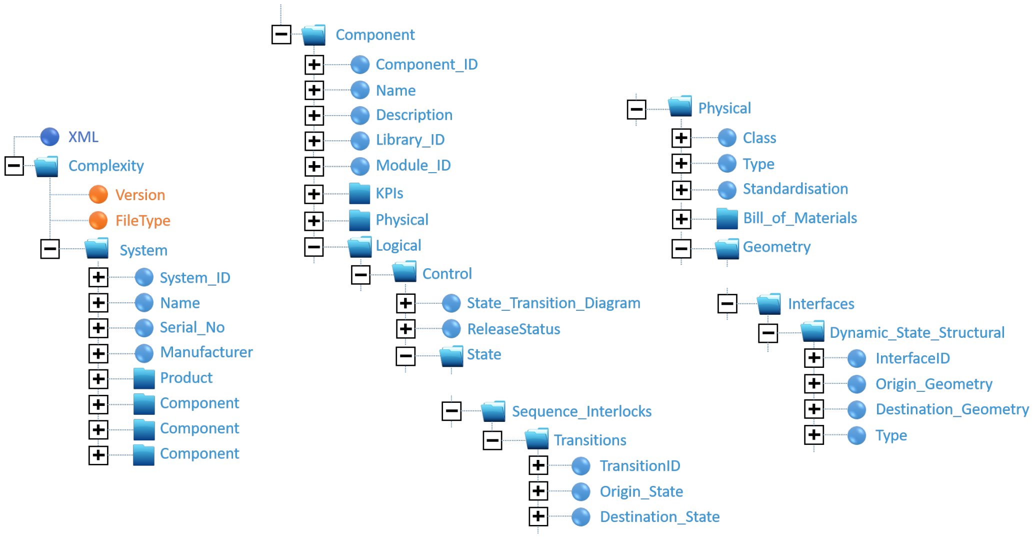

Figure 7.9: *Complexity*.XML data structure

7.4 Case studies

In this section, the proposed framework is tested using three virtual assembly system designs. In the first study, the Festo MPS system explained in the previous chapter is modelled in the vueOne toolset, and its physical complexity is re-evaluated using the proposed MATLAB application. This study aims to analyse the effects of system decomposition level on the static complexity. The second case study addresses static complexity of two different workstation configurations performing the same battery-cell assembly operation. The conceptual assembly station designs are described in terms of their structural configurations and sequence of operations to perform the required functions. Each workstation is virtually designed in the vueOne VM tool, and resulting static complexity is assessed through the developed MATLAB application. The last case study evaluates the performance of the proposed framework on the complexity assessment of large scale assembly systems. This study addresses static complexity of a vertical assembly machine performing a rotor assembly operation.

7.4.1 Festo MPS (virtual design)

Static complexity of Festo MPS was addressed in the previous chapter. In this chapter, this system is re-assessed using the proposed approach to analyse whether there is a significant deviation between the results obtained from pen-and-paper based and virtual engineering based assessments. The calculations are done for the physical system design only, as in both cases, logical system is designed by means of vueOne toolset. The physical design is intially modelled in the vueOne VE tool (**Fig. 7.10**), and then, its XML document and interface cvs. file are imported into the proposed MATLAB application. In the new physical system design, number of components is reduced from 48 to 28, and number of interfaces are dropped from 59 to 36, as the vueOne does not support modelling of interfaces such as: energy transfer and fluid flow, etc. In this manner, the tool provides a coarse representation of the system. **Figure 7.11** shows the binary adjacency matrix of the coarse system representation build by the vueOne tool.

As seen from **Figure 7.12**, there is a noticeable difference in the results of component C_1^P and interface complexity C_2^P. This is due to the finer representation of the system has large number of components and interfaces at a deeper/finer level of system decomposition. The pen-and paper method is used in a finer representation of the physical system design when compared to VueOne and hence the difference in the results provided by complexity solver and pen-and-paper based method in **Figure 7.12**. However, the topological complexity results C_3^P do not seem to differ. This is attributed to the fact that basic structure of the system remains the same beyond a level of decomposition that adequately describes and differentiates the system.

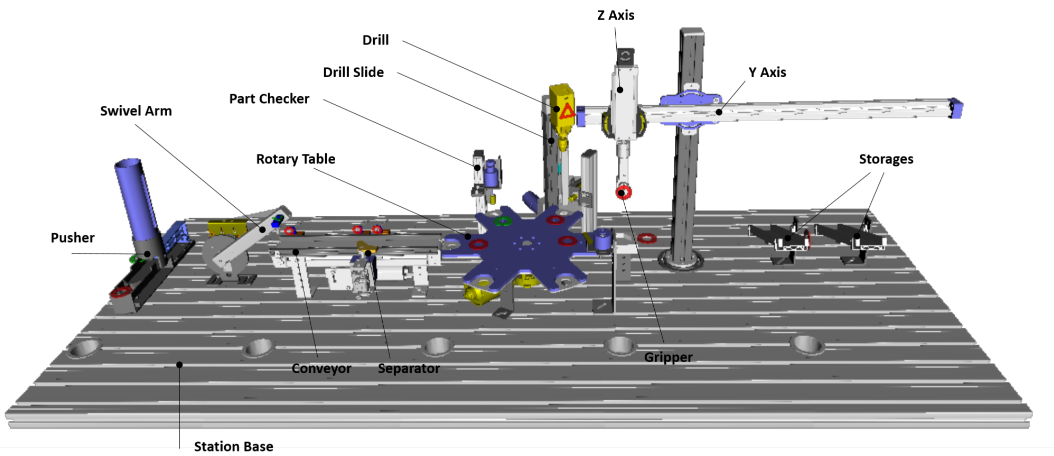

Figure 7.10: Virtual model of the Festo MPS.

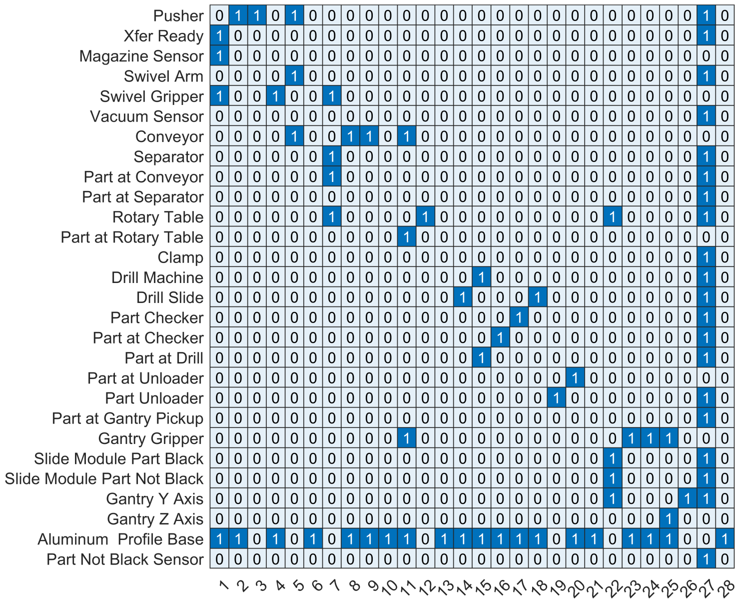

Figure 7.11: Binary DSM of the Festo MPS (Physical design).

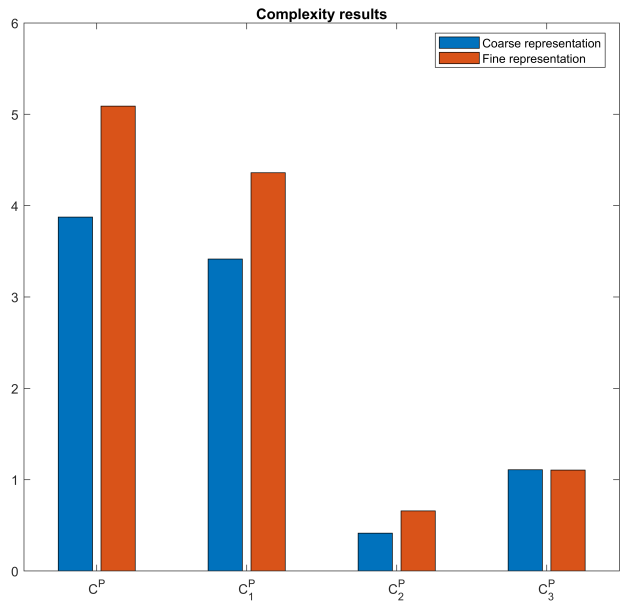

Figure 7.12: Complexity results of coarse and fine representations of Festo MPS.

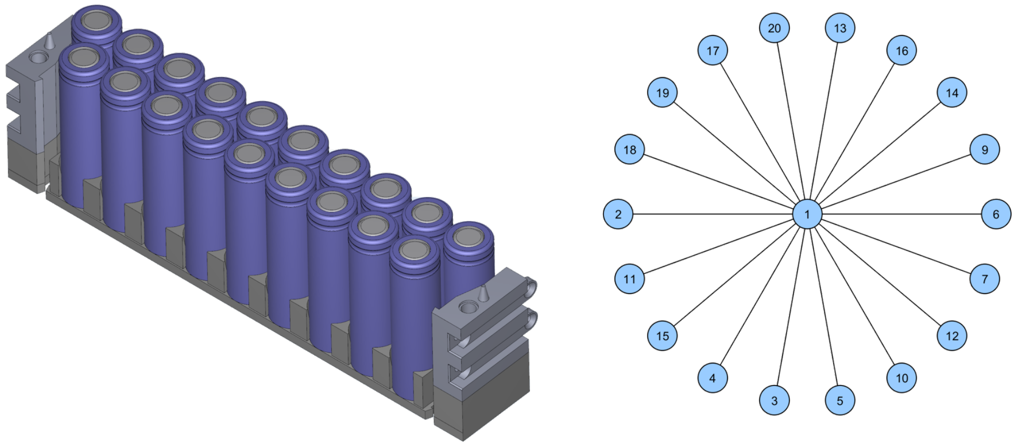

Figure 7.13: Battery cell sub-assembly (19 cylindrical cells) and its liaison diagram C^p = 5.024 C_3^p = 0.436.

7.4.2 Battery-cell assembly concepts

In this section, a pick and place workstation from a pilot production line for automotive powertrain assembly is selected to demonstrate the proposed design support framework. The workstation belongs to a full-scale automation system workbench (ASW), which is installed at Warwick Manufacturing Group to support the research and development activities of Automation Systems Group (ASG). The ASW is also used in collaboration with industrial partners, e.g. Jaguar Land Rover, etc., for demonstration of product assembly activities. This workbench is presently performing battery sub-module assembly operations as a part of the Knowledge Driven Configurable Manufacturing (KDCM) EPSRC project.

The pick and place station is expected to assemble two variants of cylindrical cells, i.e. 18650 and 26650 cylindrical cells, into corresponding bottom cell trays. There are two variants of cell trays; 11 cells and 19 cells, which will be identified by RFID tags. During the conceptual design phase, two different battery cell assembly workstation designs are virtually modelled in the vueOne VM tool. **Figure 7.13** shows CAD model of the sub-assembly and its liaison diagram. Accordingly, assembly complexity of the sub-assemblies are found as 5.024 and 3.001 for 19 cell and 11 cell variants, respectively. In both cases, the topological complexity is calculated as 0.436 indicating a highly centralised assembly structure.

In this section, static complexity of each concept design is analysed using the proposed complexity engine, and consequently, a set of recommendation is provided for possible design improvements.

7.4.2.1 The concept design *A*

In the concept design *A*, an Automated Guided Vehicle (AGV) with a tray with battery boxes feeds batteries to the workstation. A sensor located on the station indicates the presence of the pallet when it arrives at the specified location (*A1*). This actuates the pallet locator to lock the pallet in position for further operations to be performed. The robot begins to place the batteries at programmed locations on the module. The location uncertainty is considered negligible due to the sensor which detects the tag on the pallet to determine the place location for the batteries. On completion of the assembly task, the pallet locator releases the pallet, following which the pallet is conveyed to the next station. **Figure 7.14** shows the schematic layout of the concept design A.

The sequence of operation and virtual model of the concept design are developed as well as simulated in the vueOne VM tool (**Fig. 7.15**). The concept design A consists of a six axis robot, four actuators, three sensors, and three static (i.e. non-controlled) physical components, as well as eight logical components which are used to describe process behaviours of robot, actuator and sensor components. In the model, the logical description AGV component is not modelled, as its process behaviours is not attached to the overall station automation logic. The STDs of each controlled components, i.e. robot, stopper, pallet locator, gripper, conveyor, and sensors, are modelled using the component editor, and corresponding interlocks are then developed to attain the planned sequence of operation in a controlled

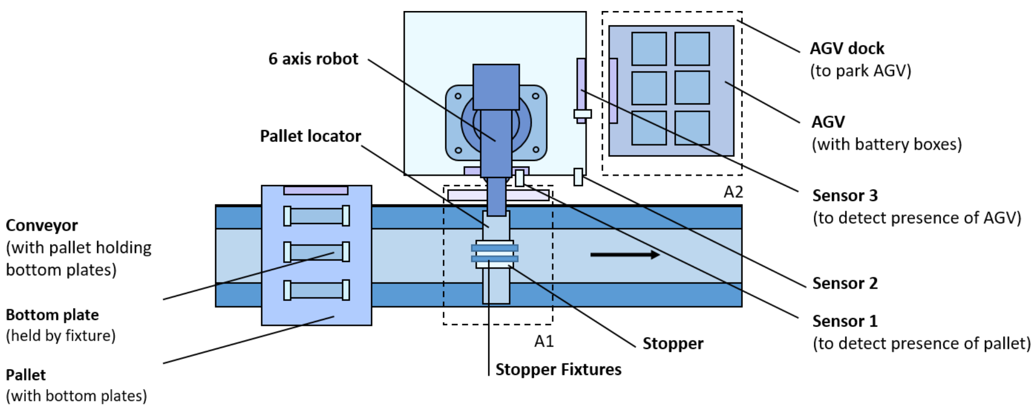

Figure 7.14: Schematic layout of the concept design A.

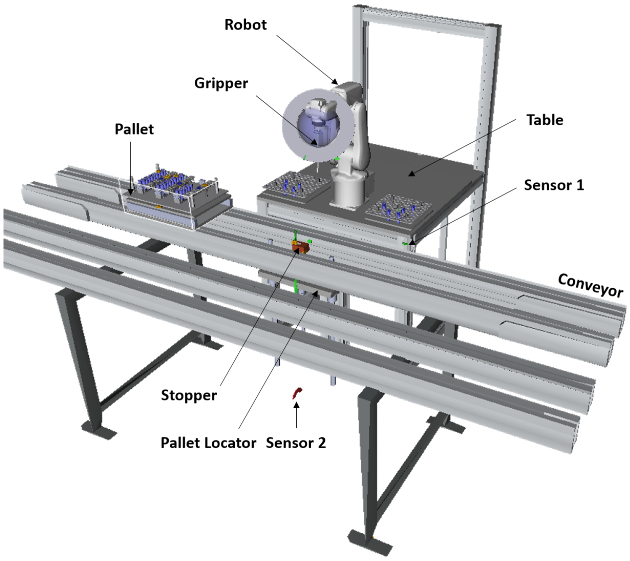

Figure 7.15: The vueOne model of the concept design A.

manner.

7.4.2.2 The concept design *B*

The second design differs from the previous one in terms of the pick and placing unit and battery feeding mechanism selections. Instead of having an AGV to feed the battery cells to the station, a feeding system comprising a conveyor with pallet is used to feed both batteries and bottom plates into the station. The pallet has a tag which is read by a sensor on the loading station and informs the station about the arrival of the pallet. This enables the pallet locator to hold the pallet at the required assembly position A. A Cartesian coordinate pick and placing unit consisting of 3 linear actuator components (i.e. X Axis, Y Axis, and Z Axis) and a dedicated two finger pneumatic gripper component, performs the assembly operation. Please note that, gripper is equipped with a special apparatus allowing it to assemble a batch of battery cell at the same time. Once the operation is finished, the assembled parts leave on the conveyor. **Figure 7.16** shows the schematic layout of the concept design B.

The concept design B is also modelled and simulated in the vueOne VM tool. The virtual model of the concept design consists of six actuators, three sensors and five static (i.e. non-controlled) physical components, and eight logical components.

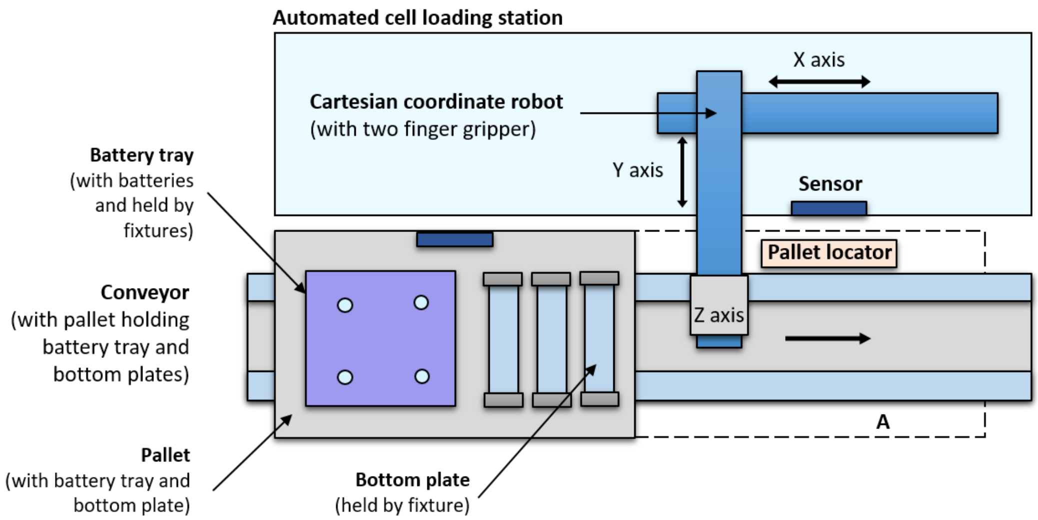

Figure 7.16: Schematic layout of the concept design B.

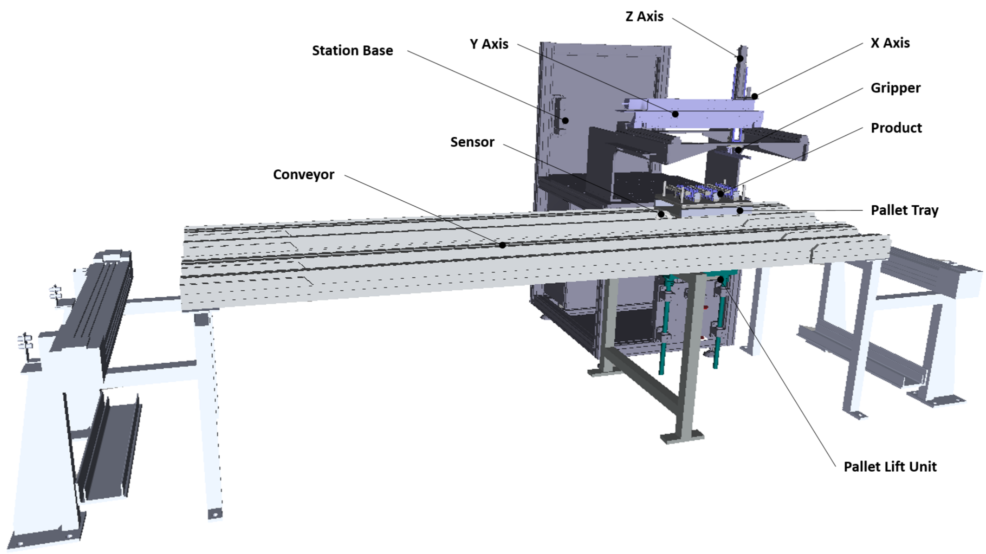

Figure 7.17: The vueOne model of the concept design B.

Please note that, both designs are developed without using a process orchestrator component. **Figure 7.17** shows the virtual model of the concept design B.

7.4.2.3 Assessment of static complexity

Static complexity of the concept designs is analysed through the developed MATLAB application. The solver reads through all components, and documents all available pair-wise interactions modelled in the vueOne VM tool. These interactions include: steady state structural, dynamic state structural, spatial connections, part transfer, event and electrical dynamics interfaces. Please note that, if at least one connection exists between two components, the application considers them connected. The representative interface factors for these connection types described above were determined by the designer, and are listed in **Table 7.2** given below. These values can be manually changed depending on the application context and size of the model from the GUI. By considering above mentioned interface types, internal block diagrams (IBDs) and binary DSM of the concept design A and B were generated as shown in **Figure 7.18** and **Figure 7.19**, respectively.

The calculation of topological complexity does not require any user input, and can be calculated solely based on the vueOne XML output. Please note that, as only mechanical and logical interfaces are included in the model, complexity should be expected increase, as more information become available at later design stages. The solver calculates the topological complexity as shown in **Figure 7.20**. From **Figure 7.19**, the topological complexities for concepts A and B look similar and are calculated around a value of 1.5, indicating a hierarchical connectivity pattern, however, a noticeable difference in complexity is visible in the logical domain. Concept design *B* evidently has higher topological complexity than Concept design *A* and this might be because of the high modularity of the pick and place system in Concept design *B*.

Component and physical complexities of the considered concept designs are depicted in **Table 7.3**. The concept *A* has higher complexity in both physical and

Table 7.2: Interface factors for battery cell assembly application.

Connection type	Interface factor, c_k
Steady state structural	0.05
Dynamic state structural	0.10
Spatial	0.07
Part transfer	0.10
Electrical dynamics	0.20
Event exchange	0.05

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Sensor 1(P)	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
Sensor 2(P)	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Stopper(P)	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Stopper Fixture(P)	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Conveyor(P)	0	1	1	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	1	0
Pallet(P)	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
Gripper(P)	0	0	0	0	0	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0
Robot(P)	0	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0
Table(P)	1	0	0	0	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0
Pallet Locator(P)	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
AGV(P)	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Sensor3(P)	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
Robot(L)	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1
Gripper(L)	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
Sensor 1(L)	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0
Sensor 2(L)	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
Stopper(L)	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Pallet Locator(L)	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0
Conveyor(L)	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
Sensor3(L)	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0

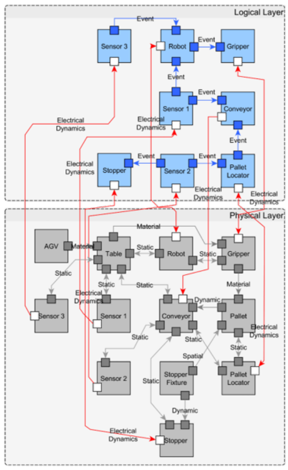

Figure 7.18: Binary DSM (*left*) and internal block diagram (*right*) of the concept design A.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Gripper(P) 1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Jaw 1(P) 2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Jaw 2(P) 3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pallet Lift Unit(P) 4	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0
Pallet Non-retun Base(P) 5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Pallet Non-retun Mech(P) 6	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
X Axis(P) 7	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Y Axis(P) 8	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
Z Axis(P) 9	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Sensor 1(P) 10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
Sensor 2(P) 11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
Sensor 3(P) 12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0
Conveyor(P) 13	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1
Pallet(P) 14	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Station Frame(P) 15	0	0	0	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Gripper(L) 16	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
Pallet Lift Unit(L) 17	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0
X Axis(L) 18	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0
Y Axis(L) 19	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0
Z Axis(L) 20	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0
Sensor 1(L) 21	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Sensor 2(L) 22	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
Sensor 3(L) 23	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1
Conveyor(L) 24	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0

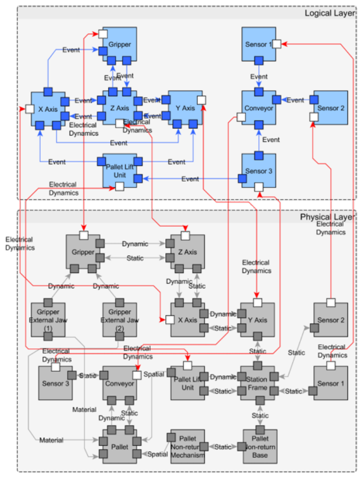

Figure 7.19: Binary DSM (*left*) and internal block diagram (*right*) of the concept design B.

logical domains. The physical complexity in the concept design *A* mainly arises from robot and AGV components. On the other hand, the physical complexities of the various components in the Concept design *B* exhibit similar values. Considering the complexity in logical domain, the robot component in The concept *A* has higher complexity than the remaining components. The robot performs pick and place operations continuosly, thereby leading to a lengthy process sequence. On

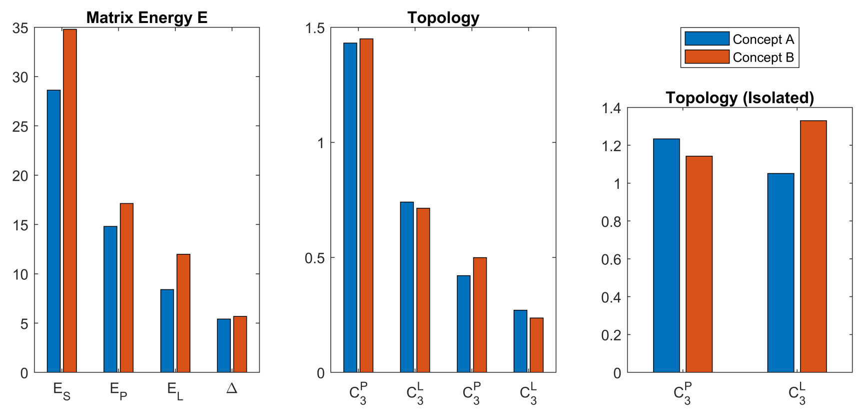

Figure 7.20: Topological complexity results of the pick and place conceptual designs.

the contrary, the pick and place system in concept B encompasses three components amongst which the complexity is distributed. As a result, there is no specific component in the Concept *B*, that has significantly high complexity value than the rest. Therefore, this approach is able to detect the source of complexity to a high level of detail which could provide useful information on critical areas of concern for reducing complexity.

Table 7.4 shows that the interface complexity of the concept design *B* in the logical domain is relatively higher than that of the concept design *A*. The reason being the high number of event interfaces and logical interlocks emanating from the

Table 7.3: Component complexities in the physical and logical domains.

	Physical components				Logical components			
	Concept A		Concept B		Concept A		Concept B	
1	Sensor 1	0.064	Gripper	0.259	Robot	0.925	Gripper	0.171
2	Sensor 2	0.064	Jaw 1	0.035	Gripper	0.150	Pallet Lift Unit	0.150
3	Sensor 3	0.064	Jaw 2	0.035	Sensor 1	0.072	X Axis	0.471
4	Stopper	0.115	Pallet Lift Unit	0.233	Sensor 2	0.072	Y Axis	0.150
5	Stopper Fixture	0.044	Pallet Non-retun Base	0.035	Sensor 3	0.072	Z Axis	0.286
6	Conveyor	0.322	Pallet Non-retun Mech	0.070	Stopper	0.150	Sensor 1	0.072
7	Pallet	0.085	X Axis	0.277	Pallet Locator	0.150	Sensor 2	0.072
8	Gripper	0.259	Y Axis	0.277	Conveyor	0.150	Sensor 3	0.072
9	Robot	0.772	Z Axis	0.277			Conveyor	0.150
10	Table	0.044	Sensor 1	0.067				
11	Pallet Locator	0.172	Sensor 2	0.067				
12	AGV	0.589	Sensor 3	0.067				
13			Conveyor	0.322				
14			Pallet	0.035				
15			Station Frame	0.035				
Total		2.594		2.091		1.741		1.596

Table 7.4: Total interface complexity (Pick and place concepts).

	Concept A	Concept B
Physical Interfaces	0.306	0.313
Logical Interfaces	0.215	0.537
Interdomain Interfaces	0.394	0.396

Table 7.5: Overall subsystems complexity (Festo MPS).

	Overall	Physical	Logical	Integrative
Concept A	5.6462	2.8205	1.8324	0.9933
Concept B	5.4934	2.3145	1.8640	1.3149

various components.

The overall complexity is calculated in **Table 7.5**, with an assumption that the physical, logical and integrative domains are equally important. From **Figure 7.21**, the areas, covered by both concepts are almost the same, indicating the same overall complexity value, however, the particular domains where the complexity is high or low differs between the two concepts. In summary, the concept design A has high component complexity with low topological complexity, whereas it is vice-versa in the concept design B. For a particular value of the overall complexity, there exists a trade-off between high topological complexity for a system of simple components and low topological complexity for complex components and interfaces.

7.4.2.4 Discussion

According to the results, both conceptual designs have similar complexity scores. The concept design A has higher component complexity score due to the selection of pick and place and feeding units, whereas the concept design B has higher topological complexity. Analysing the modularity of the above discussed concept designs, system modularity is calculated based on an approach proposed by [Holtta et al., 2005]. The approach quantifies the degree of modularity based on the average weighted decay rate of sorted singular values of the system's binary design

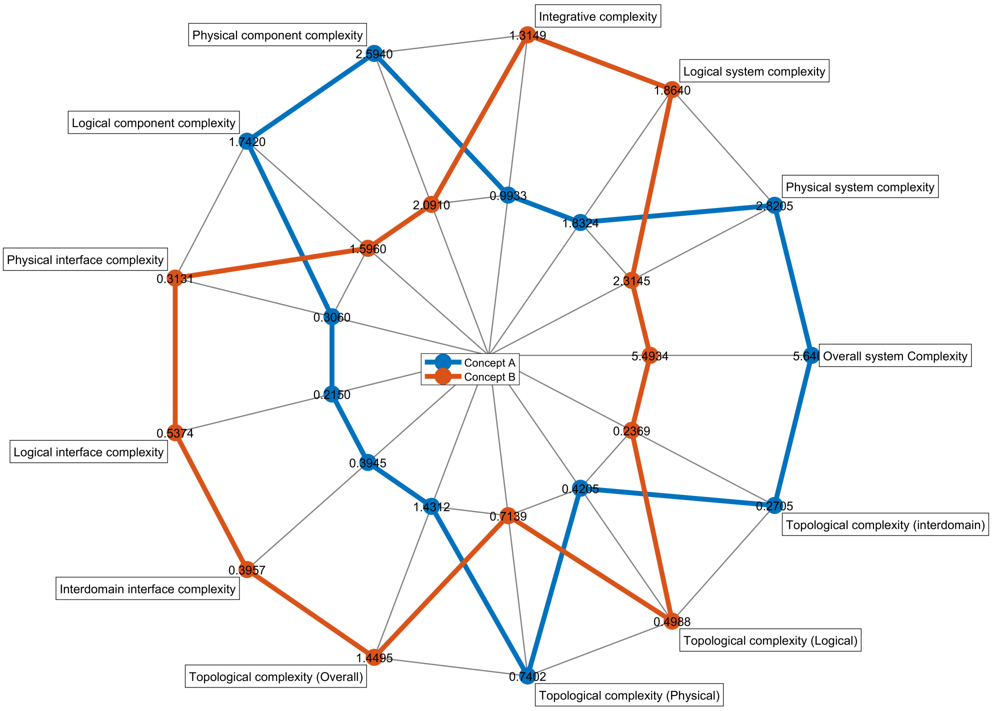

Figure 7.21: Complexity comparison of the pick and place concept designs

structure matrix.

$$SMI_{\sum DSM} = 1 - \frac{1}{N\sigma_1}\sum_{i=1}^{N-1}\sigma_i(\sigma_i - \sigma_{i+1}) \tag{7.1}$$

where, N is the total number of singular values. This index is limited between 0 and 1. The SMI closer to 1 depicts higher degree of modularity, where the connectivity information of the system is broadly distributed.

Accordingly, the degree of modularity of concept designs A and B are found as 0.8951 and 0.9104, respectively (**Figure 7.22**). The SMI metric suggests that the both concept designs are identical as the curve has a similar decreasing trend. However, the concept design *B* has slightly more modular architecture, due to the modular pick and place unit (i.e. carthesian gantry).

Figure 7.23 places the concept designs on the modularity-complexity trade-off chart. Both systems have high modularity which indicates effectively distributed total complexity across the systems. For a system with high complexity, increasing

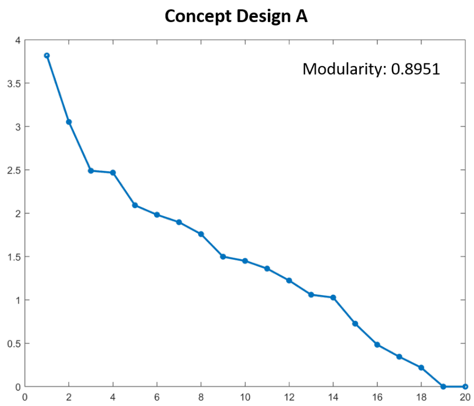

Figure 7.22: Weighted decay rate of sorted singular values of connectivity structures: *right:* concept design A, *left:* concept design B

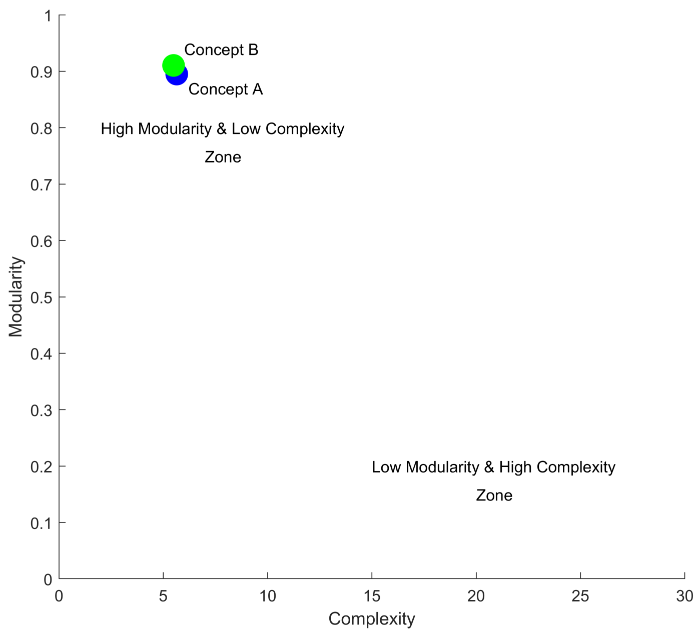

Figure 7.23: Complexity-modularity trade-off chart

the modularity could be a potential solution for managing the system since the complexity is distributed into manageable parts. Since the two concepts considered here have similar complexity and modularity, for decision making other design criteria such as cost, flexibility etc. can be considered. For example, the concept design *A*, which has a robot for performing pick and place operations has more flexibility in terms of pick and place locations, and workspace range, on the contrary, the concept design *B* is less expensive with dedicated automation.

7.4.3 Vertical assembly machine

The final test case is done to attest the working of the proposed approach for a large manufacturing system. The assembly system is designed under High Volume E-Machine Supply from the UK (HVEMS-UK) project [HVE]. The project aims to create an experimental "*make-like-production*" (MLP) facility at the campus of the University of Warwick. During this project partners participated in the investigation of manufacturing and assembly methods for the production of the electric machines. The facility included prototype machine tools and assembly systems allowing research in the area of vehicle electrification.

7.4.3.1 Configuration and process description

The vertical assembly machine is composed of:

1. Loading robot
2. Unloading robot
3. Safety door (double acting pneumatic cylinder)
4. Hub loading fixture placed at the servo press plunger (double acting pneumatic cylinder)
5. Lamination support unit (left and right) (double acting pneumatic cylinder)
6. Lamination pressing fixture (left and right) (double acting pneumatic cylinder)
7. Rotating peening plate (stepper motor)
8. Roller conveyor
9. Loading and unloading trolleys to store parts and final product
10. safety guardings.

Process requires insertion of 2 laminations on the rotor hub by vertical assembly machine. Process steps are:

1. When cycle start, safety door opens automatically
2. Loading robot places hub at the machine fixtures
3. Safety door closes and two lamination supporting units go at work position
4. Loading robot places first lamination at the supporting units
5. Orientation devices check that lamination is in correct orientation
6. Door is closed. Hub fixture moves upwards and presses lamination into the hub
7. Repeat the process 4-6 for second lamination
8. After second lamination was pressed at the hub, peeing process starts automatically: top fixtures moves aside. Press plunger moves upwards with product to execute peening
9. Peening plate rotates, peening process cycle repeats three times
10. When peening cycle is complete, hub with laminations (final assembled product) goes down to the conveyer
11. Unloading robot takes assembled product from conveyor and places it at the trolley.

Final product assembly is achived by pressing togehter the three parts: hub and 2 laminations. The product is the main part of the rotor assembly which will be further used in electric motor production for the automotive industry.

7.4.3.2 Assessment of static complexity

Vertical assembly machine is modelled in the vueOne VM tool using 33 physical and 15 logical components. **Figure 7.24** shows the virtual model of the machine. The system has been intutively identified as being large and complex by the system engineers. Static complexity of the vertical assembly machine is analysed through

Table 7.6: Interface factors for vertical assembly machine.

Connection type	Interface factor, c_k
Static structural	0.05
Dynamic translational	0.10
Dynamic rotational	0.10
Spatial	0.07
Part transfer	0.10
Electrical dynamics	0.20
Event exchange	0.05

the developed MATLAB application. The solver read through all components, and documents all available pair-wise interactions modelled in the vueOne VM tool. These interactions include: static structural connections, translational kinematics, rotational kinematics, spatial connections, part transfer, event and electrical dynamics interfaces. The representative interface factors for these connection types described above were determined by the designer, and are listed in **Table 7.6** given below. These values can be manually changed depending on the application con-

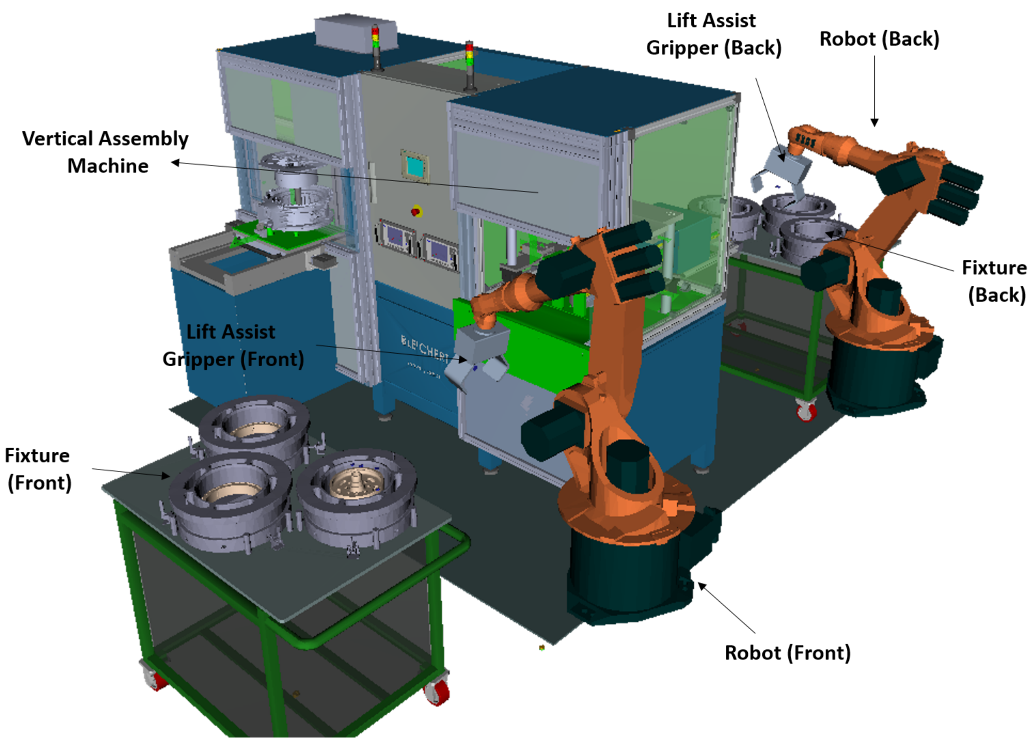

Figure 7.24: The vueOne model of the vertical assembly machine

text and size of the model from the GUI. By considering above mentioned interface types, internal block diagrams (IBDs) and binary DSM of the vertical assembly machine were generated as shown in **Figure 7.25** and **Figure 7.26**, respectively.

Complexity C_3^S of the overall vertical machine structure is found as 1.439 with a graph energy $E_{[MDM]^S}$ of 69.079 ($E_{[DSM]^P}$=40.9019, $E_{[DSM]^L}$=17.6954, and Δ=10.4818). This indicates a hieararchical connectivity for the overall system architecture. The contribution of physical, logical, and interlayer topologies to the overall topological complexity is found to be 0.852, 0.369, 0.218, respectively.

Table 7.7 shows the complexity estimations of physical and logical system components. It is important to note, the logical architecture of the vertical assembly machine is partially modelled (i.e. does not include assembly process sequences), therefore, the total component complexity in the logical domain is found to be relatively smaller than its physical counterpart. Similarly, total interface complexity of the vertical assembly machine for physical, logical and inter-domain interfaces are recorded as 1.181, 0.818, and 1.008, respectively.

As the final step, overall complexity of the system is measured by assuming that the physical, logical and integrative domains are equally important (w^P = w^L = w^Δ=1). Accordingly, the overall system complexity is recorded as 13.215. Please note that, it is expected to have an increase in the static complexity of the vertical assembly machine once the missing information is fully captured. **Figure 7.27** compares the static complexity results of the vertical assembly machine with the previously analysed battery cell pick and place workstation concepts' complexities. The results clearly reflect the intuitive opinions of the system designers on the static complexity of these assembly systems.

7.5 Subjective validation

In this section, the suitability of the proposed complexity assessment framework is considered for the identification of highly complex assembly workstations during conceptual design phases. Since no accurate information is available regarding the 'absolute' complexity of the production systems i.e. system development effort or time, it was decided to ask system engineers to jointly 'nominate' both the most complex and the simplest workstation designs. The three participants were drawn from ASG at University of Warwick and were currently serving as system

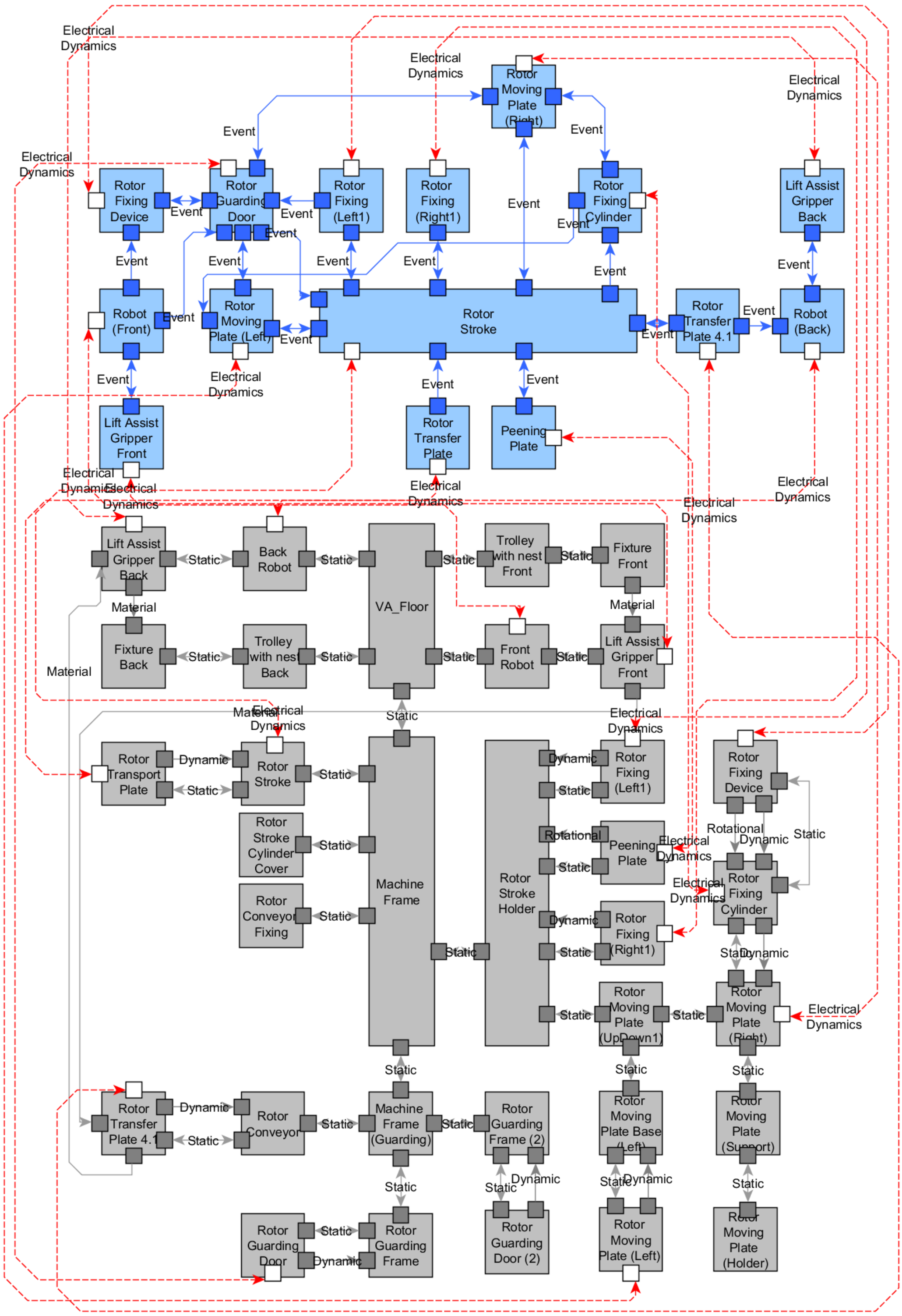

Figure 7.25: Internal block diagram of the vertical assembly machine

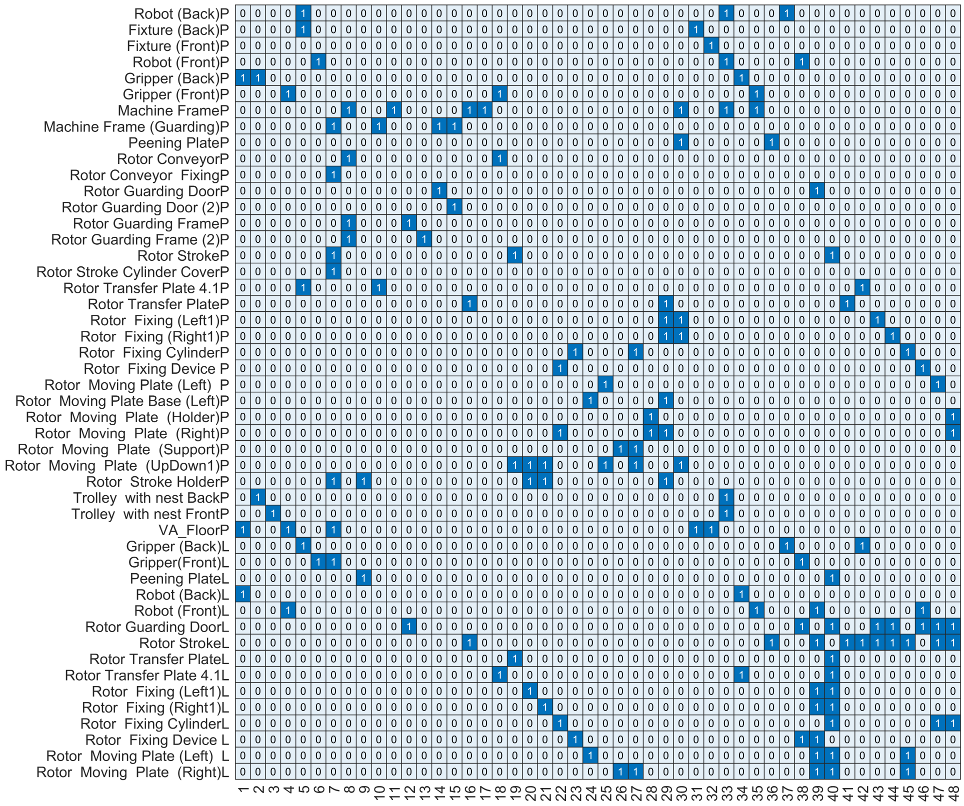

Figure 7.26: Multi-domain matrix of the vertical assembly machine

engineers on a variety of large, high-profile projects. The participants were asked to subjectively assess design complexity of twenty-three virtual assembly systems modelled in the vueOne VM toolset. All stations are component-based assembly automation systems performing operations in automotive power-train and electric battery assembly applications. Accordingly, thirteen of them were deemed of LOW complexity and ten HIGH by the system engineers. We subsequently plotted out the resulting scores obtained from MATLAB application for all twenty-three workstations, and their subjective complexity label LOW to HIGH (**Figure 7.28**). Please note that, in the calculations, only overall system complexity scores were considered. Based on the standard t-test, that the average score for LOW stations, 5.07 (with SD of 2.35), differs significantly from the average of the HIGH stations, being 10.93 (SD of 3.10).

A statistical model that will find the correlation between the quantitative scores achieved through the design support framework and the 'perceived' level of complexity, is proposed. Since, there is only two (dependent) responses (HIGH or LOW), a linear regression is not appropriate for a statisical model. There exists, however, a model, called 'Logistic Regression' or 'Logit', that will calculate the probability that the resulting score of the assembly system design indicates either HIGH or LOW complexity. Note that the values 0 and 1 can be assigned either

Table 7.7: Component complexities in the physical and logical domains (vertical assembly).

	Physical components		Logical components	
1	Robot (Back)	0.772	Gripper (Back)	0.112
2	Fixture (Back)	0.142	Gripper(Front)	0.112
3	Fixture (Front)	0.142	Peening Plate	0.078
4	Robot (Front)	0.772	Robot (Back)	0.342
5	Gripper (Back)	0.259	Robot (Front)	0.342
6	Gripper (Front)	0.259	Rotor Guarding Door	0.078
7	Machine Frame	0.105	Rotor Stroke	0.342
8	Machine Frame (Guarding)	0.105	Rotor Transfer Plate	0.176
9	Peening Plate	0.172	Rotor Transfer Plate 4.1	0.176
10	Rotor Conveyor	0.135	Rotor Fixing (Left1)	0.078
11	Rotor Conveyor Fixing	0.172	Rotor Fixing (Right1)	0.078
12	Rotor Guarding Door	0.178	Rotor Fixing Cylinder	0.078
13	Rotor Guarding Door (2)	0.178	Rotor Fixing Device	0.078
14	Rotor Guarding Frame	0.111	Rotor Moving Plate (Left)	0.078
15	Rotor Guarding Frame (2)	0.111	Rotor Moving Plate (Right)	0.078
16	Rotor Stroke	0.467		
17	Rotor Stroke Cylinder Cover	0.070		
18	Rotor Transfer Plate 4.1	0.356		
19	Rotor Transfer Plate	0.356		
20	Rotor Fixing (Left1)	0.222		
21	Rotor Fixing (Right1)	0.222		
22	Rotor Fixing Cylinder	0.178		
23	Rotor Fixing Device	0.470		
24	Rotor Moving Plate (Left)	0.178		
25	Rotor Moving Plate Base (Left)	0.070		
26	Rotor Moving Plate (Holder)	0.035		
27	Rotor Moving Plate (Right)	0.178		
28	Rotor Moving Plate (Support)	0.035		
29	Rotor Moving Plate (UpDown1)	0.035		
30	Rotor Stroke Holder	0.035		
31	Trolley with nest Back	0.070		
32	Trolley with nest Front	0.070		
33	VA Floor	0.000		
Total		6.661		2.227

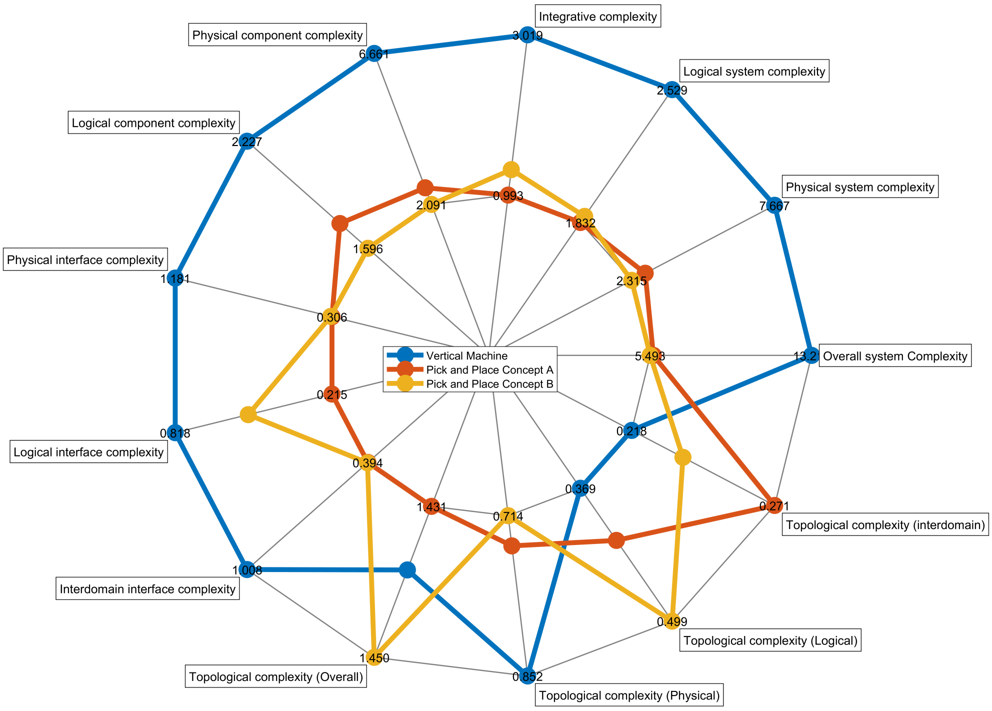

Figure 7.27: Comparison of vertical assembly machine complexity with pick and placing concept designs

way, i.e. HIGH = 1, LOW= 0 or vice versa. The actual calculations were done with HIGH = 1, LOW= 0. Since we assigned the value 1 to high, the model will start with a probability of 0 of LOW through the constant, and the scores will reduce this probability because the weight is positive. Accordingly, the LOGIT model looks like this:

$$P_{LOGIT}(High) = \frac{e^{-5.73+0.706C^S}}{1+e^{-5.73+0.706C^S}} \tag{7.2}$$

Table 7.8 contains the numerical results, as output by Minitab. The model succeeds in classifying the cases 87 percent of the time correctly. **Figure 7.29** shows the binary fitted line plot of logistic regression model. Although the subjective validation is carried out using a limited sample size, these results show us that the proposed approach can be used as a valid complexity indicator especially useful in

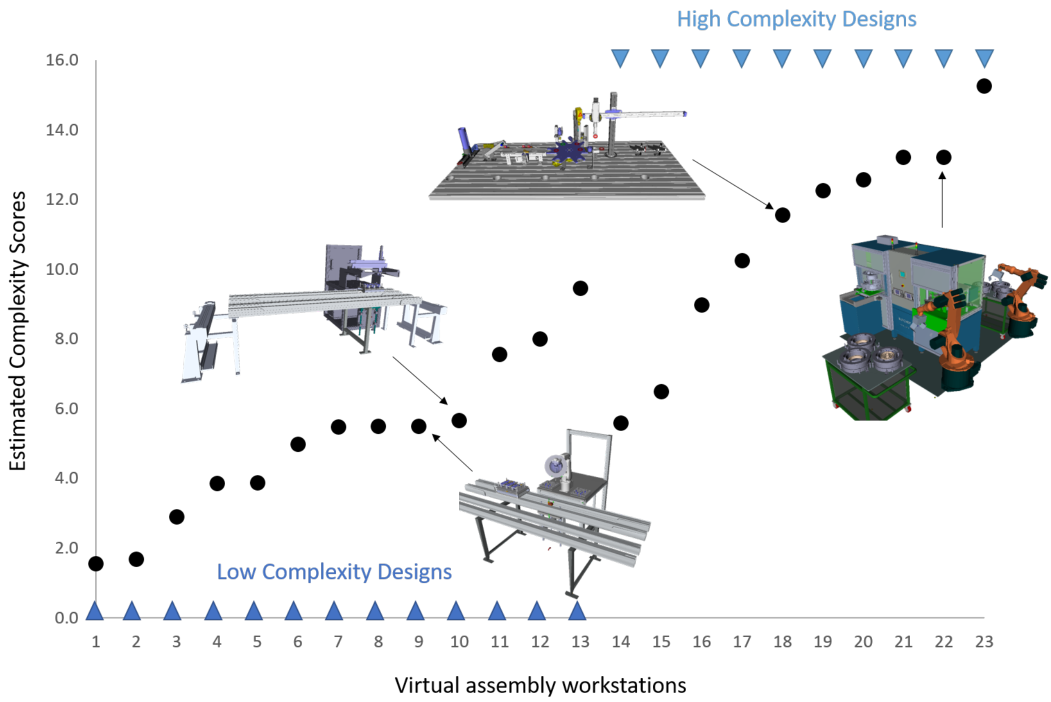

Figure 7.28: Model score compared to subjective complexity for all workstations.

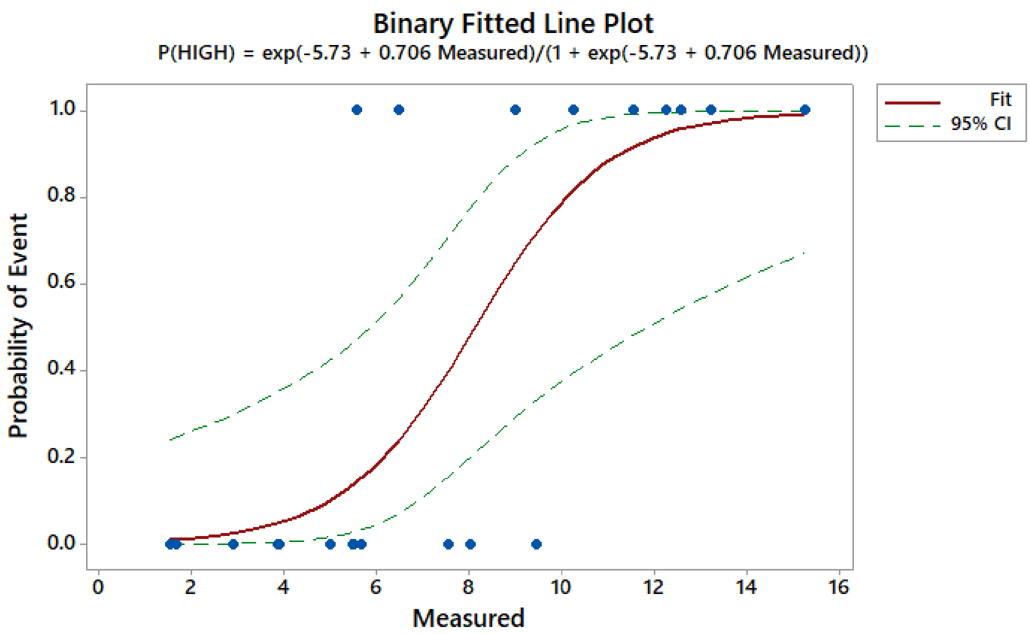

Figure 7.29: Fitted line plot for LOGIT model

Table 7.8: Statistical fit results for LOGIT model (The cut value is 0.5).

Observed	Predicted		Percentage correct
	High	Low	
High	8	0	80
Low	0	15	86.7

early design phases.

Using this information, further research should look into the cases were the subjective labels are not match the model scores to assess whether the subjective label is wrong or not. In the former case, this will enhance the value and validity of the model, and yield information about the subjective reasoning that led to the wrong classification. In the latter case, the mathematical model will be refined using a collection of historical data to ensure that complexity measurement and assembly systems engineering characteristics can be properly correlated. The scores from the model could be used as independent variable for researching the impact of complexity on both direct and indirect costs, and on the subjective interpretation of complexity.

7.6 Discussion

In this chapter, a virtual engineering based approach is proposed to assess the static complexity of virtually designed assembly systems.

The benefits of the proposed approach are threefold, i) complexity assessment can be performed in the early design phase, where any change in design and its corresponding change in complexity can be assessed with minimal implications on cost and time ii) the complexity inclusive design support approach is automated and hence eliminates the laborious manual work associated with the existing approaches iii) complexity assessment performed encompasses different domains, thereby allowing the detection of the exact source of complexity with a great level of detail which enables identification of critical points or aspects which could help to reduce complexity at the design stage.

Furthermore, to verify the benefits of the proposed approach, a workgroup, comprising of system engineers, was established and their opinions and intuitive knowledge on the complexities of different systems was assessed. A comparison of this value with the values obtained by the approach was reviewed by the work group and was found to be broadly in accordance with their results.

7.7 Chapter summary

In this chapter, the previously presented complexity assessment approach is integrated into a virtual system design and development software to support early life-cycle phases of component-based assembly automation systems. The approach was demonstrated using a three different case studies with varying degree of complexity. Furthermore, a statistical model was developed to validate the complexity-inclusive design support framework.

Chapter 8

Conclusions and future work

This chapter concludes the research work presented in this thesis through summarising the research contributions, and pointing out possible future research work.

8.1 Achievement of research objectives

To fulfil the research aim, four main research objectives were described in the first Chapter. This section highlights the achievements towards these objectives.

- **Objective 1: Identify and examine the existing concepts and approaches related to the characterisation of manufacturing systems complexity**

  To this end, the drivers of complexity and typical symptoms of complexity in manufacturing systems design were identified. A comprehensive review of studies published within the last two decades to assess manufacturing system complexity was carried out, and a taxonomy which comprehensively captures all complexity assessment methodologies that the author has been able to identify from a structured literature review was presented.

  The conclusion drawn from this part of the study is that there are many complexity assessment methods, yet there is limited work on how those methods can be translated into a practical solution that can be used to support engineering tasks. Accordingly, the approaches that examine complexity during the operational phases are often costly; they require large datasets collected by on-site observations/measurements and often analysed using expert systems. On the other hand, approaches measuring complexity at the design stages of manufacturing systems are less successful as the large amount of data required are not available at this point of a systems life-cycle. Also, traditional approaches on complexity measurement at conceptual design phases are often pen-and-paper based methods and heavily rely on empirical knowledge or domain specific expertise. Consequently, an assessment of complexity at the design phase remains tedious and time consuming, as managers/engineers require practical efficient methods for its measurement.

- **Objective 2: Development of an adaptable, cohesive and concise definition of static complexity for assembly products and systems**

  As a consequence of the research gap identified, the second research objective was defined so as to develop a mathematical definition of static complexity, which can be used in identifying and verifying critical design parameters of assembly systems and products in a quantitative and repeatable fashion. To this end, a mathematical model based on an analogy derived from molecular physics allowing us to map static complexity to the overall system development effort was adopted. The approach is first applied to the industrial product assemblies. Accordingly, assembly complexity is defined as a combination of both the complexity of product entities and their topological connections. In this model, complexity of product entities (*i.e.* components and liaisons) is defined as the degree to which the entity comprises structural characteristics that lead to challenges during handling or fitting operations. The characterisation of entity complexities are carried out based on widely used DFA principles.

  The mathematical model is then used to define static complexity of component-based assembly systems. In the approach static system complexity is defined as a function of: i) complexity of isolated system entities (i.e. logical and physical components and interfaces), and ii) complexity arising due to the systems connectivity pattern between those entities. In this context, complexity of isolated system entities was defined as the relative development and management effort/cost of the entity itself, whereas complexity of the systems structural connectivity pattern was estimated through the matrix energy of product design structure matrices. The applications of the approach showed that the approach is mathematically rigorous and lends objectivity to complexity analysis.

- **Objective 3: Development of a framework that brings together the mathematical definition of complexity and its practical applicability to real-world system development**

  The third research objective was to translate the above complexity assessment approach (itself derived from the theoretical complexity model) into a set of engineering methods and corresponding design support tools that can be integrated to an existing Virtual Process Planning software solution, namely vueOne Engineering tool set, so that engineers can analyse and further optimise static complexity within this virtual design environment. Towards this aim, a novel proactive design support framework, in which virtual engineering data sets are streamlined as an input to a complexity assessment model (objective 2) was proposed.

  The capabilities of the proposed framework was demonstrated by developing a stand-alone complexity assessment software tool and its conceptual integration into the existing vueOne virtual environment. The industrial case studies showed that the proposed approach makes complexity assessment possible through the collection of structured data generated during the virtual engineering phase. The full integration of the tool is expected to provide designers with better insights of static complexity, where most benefits can be achieved with minimum system development effort, risk and disruption.

8.2 Key research contributions

This study has made the following original contributions to the field of complexity modelling and manufacturing systems engineering:

- A critical and comprehensive review of literature on complexity in production systems and recommendations for future work along with extension of existing classification schemes by inclusion of complexity management which is greatly beneficial for researchers is provided.

- A complexity assessment model focusing on assembly complexity of industrial products was developed. The proposed approach solely requires physical design information and thus, can be considered as more practical, especially for initial design stages, than the approaches requiring real production data

- A complexity assessment model focusing on static (i.e. structural) complexity of assembly systems was developed. The approach is mathematically rigorous, and assesses static complexity at a high resolution over a broad spectrum ranging from topological complexity to physical and logical system components.

- A set of design support functionalities, derived from the developed complexity model and assessment methods, and implemented as part of virtual engineering tools typically used in industry during early phase of manufacturing system engineering. This allows designers to become aware of critical points in manufacturing system design which could be critical in terms of design reliability. In particular, this allows designers to avoid over-complex solutions where possible.

8.3 Research benefits

This study is a systematic attempt to bridge the current gap between theoretical definitions of complexity and their practical applicability to real-world manufacturing system engineering. Virtual engineering is used as an enabling technology to implement complexity evaluation model and framework, in a software solution module that integrate with virtual engineering tools (in this case vueOne VM toolset) typically used to support early engineering phases. Based on the industrial case studies and a series of expert evaluations, a number of potential benefits of the proposed study were identified.

- An assessment of complexity can be made during the very early design phases so that those system designs deemed excessively complex can be flagged and optimised.

- Assessment of complexity is automated, and integrated into a virtual engineering tool through the systemic complexity definition and structured data collection, resulting in reduced measurement efforts and better accuracy.

- The proposed measure is multi-dimensional allowing us to better comprehend the root causes of complexity than the methods with single complexity value.
- Complexity can be used as an additional design performance indicator leading to increased flexibility in decision-making processes. Also, the proposed framework can be used in multi-objective design optimisation studies.
- The complexity model and approach to complexity assessment is systemic, and can therefore be applied to other systems and/or domains of manufacturing systems engineering.

8.4 Future Work

Although this research has successfully and comprehensively addressed the objectives of the thesis, there are still some challenges and associated limitations to overcome before this framework can be deployed reliably in real-world manufacturing system design projects. It is envisioned that further developments of the framework could be made towards the following additional objectives.

- **A comprehensive empirical validation through the collection of engineering data**

  The first limitation noted of this study is the lack of engineering data describing a reasonably wide and diversified spectrum of manufacturing system designs, which is required for the comprehensive validation of the theoretical model. This has led to define some of the parameters used to achieve complexity measures, in a subjective manner. In this study, simple ball-and-stick assembly experiments and expert views are employed to verify the proposed approach. However, a collection of historical data is required to calibrate and validate the model and to ensure that complexity measurement and manufacturing systems engineering characteristics can be properly correlated.

  As part of the future work, the plan is to build a structured database of information, collected from real manufacturing system design and development projects. Then, a statistical model will be developed and continuously updated using data-analytics and the stored data. A high fidelity statistical model will allow us to better comprehend and interpret the relationship between complexity and system development/management effort, ultimately leading us to develop better complexity management practices. Moreover, the real production data will also be used to achieve a statistical model correlating products' assembly complexity and their assembly systems.

- **A data driven estimation of component and interface complexities** In the proposed study, complexity of the system entities (i.e. physical and logical components and interfaces) was assessed using a heuristic-based method. Although the proposed method provides the ability to automate the process of feeding data into the model, which translate in a level of usefulness, the initial complexity calculations of novel components and interfaces can be time-consuming as the method relies solely based on subjective assessments.

  As a part of the future work, it is planned to replace the heuristics based methodology with a more data driven approach, where available engineering data about individual system elements, will be collected and then transformed into statistically valid causal models. Suitable connectivity needed for the reported gap can be fulfilled through the Industry 4.0 viewpoint. Collecting real time life-cycle parameters (i.e. mean time between failures, mean time to repair, etc.) of system components during the operation phase in a structured manner, allows us to better predict component complexities stored in the virtual component library. Note that, subjectivity in complexity assessment should not be fully abandoned as it is an important source of information, especially for cases where enough data is not exist, e.g. novel designs, etc.

- **Enriching the capabilities of vueOne virtual manufacturing tool** The virtual engineering software solution to which the complexity assessment approach was linked, has limitations.

  A major limitation associated with the vueOne tool is the deficient level of detail in modelling physical system interfaces. As an example, physical interfaces such as: electrical energy transfer, fluid flows, etc., are not modelled in the current tool, as the tool itself was not developed for such design domains. As part of the future work, a semantic-based interface modelling engine will be developed to increase the accuracy of output from the proposed framework. This engine will automatically generate the non-user defined interface information based on a series of semantic rules, therefore updating the missing interfaces although they are not modelled within the simulation.

Another limitation of the tool is associated with the modelling of logical component definitions. Currently, vueOne only facilitates sequential representations of STDs with XOR branches, thereby reducing the flexibility with which designers can model STDs. The capability to model STDs with parallelism, iterations, and nesting could potentially increase the flexibility in STD designs, by this means, allowing differentiation of the alternate designs with regard to their static complexity.

- **The integration of the proposed framework into a PLM database to realise complexity-inclusive design selections** The proposed complexity assessment framework can also be used in automatic selection of feasible system configurations. Information such as required functionality, maximum cycle time, flexibility, scalability, etc. are envisioned to be input into veOne tool. The use of a Product Life-Cycle Management (PLM) database in conjunction with the virtual system design tool, will allow the automatic generation of several alternate designs that meet the above-mentioned criteria, subsequently creating a design space of valid architectures. Consequently, an optimiser could be used to compare these designs with the support of information stored in a database, thereby providing an optimum solution that meets the requirements.

Bibliography

Delmia,2018. https://www.3ds.com/products-services/delmia/. Accessed: 2018-11-11.

Hvems,2018. https://hvm.catapult.org.uk/impact/case-studies/investigating-new-manufacturing-and-assembly-processes-for-automotive-traction-electric-machines/hvems/. Accessed: 2018-11-11.

Andres G Abad. *Modeling and analysis of process complexity and performance in mixed model assembly systems*. PhD thesis, University of Michigan, 2010.

Andres G Abad and Jionghua Jin. Complexity metrics for mixed model manufacturing systems based on information entropy. *Int. J. Information and Decision Sciences J. Information and Decision Sciences*, 3(4):313–334, 2011.

Festo AG. Linear gantries exct specification sheet, 2017.

B Ahmad. A component-based virtual engineering approach to PLC code generation for automation systems. 2014.

Mus'ab H Ahmad, Bilal Ahmad, Daniel Vera, and Robert Harrison. An Innovative Energy Predictive Process Planning Tool for Assembly Automation Systems. *Iecon 2015*, (November):5184–5190, 2015.

Mussawar Ahmad, Bugra Alkan, Bilal Ahmad, Daniel Vera, Robert Harrison, James Meredith, and Axel Bindel. The use of a complexity model to facilitate in the selection of a fuel cell assembly sequence. *Procedia CIRP*, 44:169–174, 2016a.

Mussawar Ahmad, Bugra Alkan, Bilal Ahmad, Daniel Vera, James Meredith, Axel Bindel, and Robert Harrison. Hydrogen fuel cell pick and place assembly systems : Heuristic evaluation of reconfigurability and suitability. *CIRP Annals - Manufacturing Systems*, 00, 2016b.

Miguel D. Alfaro and Juan M. Sepulveda. Chaotic behavior in manufacturing systems. *International Journal of Production Economics*, 101(1 SPEC. ISS.):150–158, 2006. ISSN 09255273. doi: 10.1016/j.ijpe.2005.05.012.

B Alkan, D Vera, M Ahmad, B Ahmad, and R Harrison. Design evaluation of automated manufacturing processes based on complexity of control logic. *26th CIRP Design Conference*, 50:141–146, 2016a. ISSN 22128271. doi: 10.1016/j.procir.2016.05.031.

Bugra Alkan, Daniel Vera, Mussawar Ahmad, Bilal Ahmad, and Robert Harrison. A Lightweight Approach for Human Factor Assessment in Virtual Assembly Designs: An Evaluation Model for Postural Risk and Metabolic Workload. In *Procedia CIRP*, volume 44, pages 26–31, 2016b. doi: 10.1016/j.procir.2016.02.115.

Bugra Alkan, Daniel Vera, Mussawar Ahmad, Bilal Ahmad, and Robert Harrison. A Model for Complexity Assessment in Manual Assembly Operations Through Predetermined Motion Time Systems. *Procedia CIRP*, 44:429–434, 2016c. ISSN 22128271. doi: 10.1016/j.procir.2016.02.111.

Bugra Alkan, Daniel A Vera, Mussawar Ahmad, Bilal Ahmad, and Robert Harrison. Complexity in manufacturing systems and its measures: a literature review. *European Journal of Industrial Engineering*, 12(1):116–150, 2018.

Pierpaolo Andriani. Evolutionary dynamics of industrial clusters. *Complex systems and evolutionary perspectives on organisations: The application of complexity theory to organisations*, 2003:127–145, 2003.

S S Asan. *A methodology based on theory of constraints' thinking processes for managing complexity in the supply chain*. PhD thesis, University of Berlin, 2009.

W. Ross Ashby. *An introduction to cybernetics*. Chapman & Hall LTD, London, 1956.

Steven Ashley. Cutting costs and time with dfma. *Mechanical Engineering*, 117 (3):74, 1995.

F M Asl, A G Ulsoy, and Y Koren. Dynamic Modeling and Stability of the Reconfiguration of Manufacturing Systems A Dynamic Systems Analogy and Simulation. *Prooceeding of the 2000 japan.USA flexible Automation Conference*, (C), 2000.

F Avram, D Bertsimas, and M Ricard. Fluid models of sequencing problems in open queueing networks; an optimal control approach. *Institute for Mathematics and its Applications*, 1995.

Sameh Nozhy Samy Badrous. *Complexity of products and their assembly systems*. PhD thesis, University of Windsor, 2011a.

Sameh Nozhy Samy Badrous. *Complexity of products and their assembly systems*. PhD thesis, University of Windsor (Canada), 2011b.

Y Bar-Yam. *Dynamics of complex systems*. 1997. ISBN 0201557487.

Z M Bi, Lihui Wang, and Sherman Y T Lang. Current status of reconfigurable assembly systems. *International Journal of Manufacturing Research*, 2(3):303–328, 2007. ISSN 1750-0591. doi: 10.1504/IJMR.2007.014727.

Jaap Bloem, Menno van Doorn, Sander Duivestein, David Excoffier, René Maas, and Erik van Ommeren. The Fourth Industrial Revolution. *VINT research report*, pages 1–39, 2014.

G Boothroyd and L Alting. Design for assembly and disassembly. *CIRP Annals-Manufacturing Technology*, 41(2):625–636, 1992.

Geoffrey Boothroyd and Peter Dewhurst. Product design for assembly handbook. *Boothroyd and Dewhurst Inc, Wakefield R*, 1987.

Tyson R Browning. Applying the design structure matrix to system decomposition and integration problems: a review and new directions. *IEEE Transactions on Engineering management*, 48(3):292–306, 2001.

A Calinescu, J Efstathiou, J Schirn, and J Bermejo. Applying and assessing two methods for measuring complexity in manufacturing. *Journal of the Operational Research Society*, 49(7):723–733, 1998. ISSN 0160-5682. doi: 10.1057/palgrave.jors.2600554.

A Calinescu, J Efstathiou, S Sivadasan, J Schirn, and Huaccho Huatuco. Complexity in Manufacturing: An Information Theoretic Approach. *Proceedings of the International Conference on Complexity and Complex Systems in Industry*, pages 30–44, 2000.

Anisoara Calinescu. Manufacturing Complexity : An Integrative Information-Theoretic Approach. 2002a.

Anisoara Calinescu. *Manufacturing complexity: An integrative information-theoretic approach*. PhD thesis, University of Oxford, 2002b.

Marco Cantamessa. The manufacturing system as a complex artifact. *Robotics and Computer-Integrated Manufacturing*, 14(5-6):403–414, oct 1998. ISSN 07365845. doi: 10.1016/S0736-5845(98)00016-7.

J. Casti. Connectivity, Complexity and Catastrophe in Large-scale Systems. *International Series on Applied Systems Analysis*, 7:203, 1979. ISSN 03772217. doi: 10.1016/0377-2217(80)90118-6.

Vincent Chan and Filippo A Salustri. Dfa: The lucas method. *Ryerson University*, 2005.

M Chapman and D Solomon. The relationship of cyclomatic complexity, essential complexity and error rates. In *Proceedings of the NASA Software Assurance Symposium, Coolfont Resort and Conference Center in Berkley Springs, West Virginia*, 2002.

Malarvizhi Kaniappan Chinnathai, Bugra Alkan, and Robert Harrison. Convertibility evaluation of automated assembly system designs for high variety production. *Procedia CIRP*, 60:74–79, 2017.

Magdalena K Chmarra, Andres A Alvarez Cabrera, Thom van Beek, Valentina D'Amelio, Mustafa S Erden, and Tetsuo Tomiyama. Revisiting the divide and conquer strategy to deal with complexity in product design. In *Mechtronic and Embedded Systems and Applications, 2008. MESA 2008. IEEE/ASME International Conference on*, pages 393–398. IEEE, 2008.

S. Cho, R. Alamoudi, and S. Asfour. Interaction-based complexity measure of manufacturing systems using information entropy. *International Journal of Computer Integrated Manufacturing*, 22(10):909–922, 2009a. ISSN 0951-192X. doi: 10.1080/09511920902951393.

S Cho, R Alamoudi, and S Asfour. Interaction-based complexity measure of manufacturing systems using information entropy. *International Journal of Computer Integrated Manufacturing*, 22(10):909–922, 2009b.

ACK Choi, DSK Chan, and AMF Yuen. Application of virtual assembly tools for improving product design. *The International Journal of Advanced Manufacturing Technology*, 19(5):377–383, 2002.

G Chryssolouris. Manufacturing systems: theory and practice. 2013a. doi: 10.1007/0-387-28431-1.

G. Chryssolouris, N. Giannelos, N. Papakostas, and D. Mourtzis. Chaos Theory in Production Scheduling. *CIRP Annals - Manufacturing Technology*, 53:381–383, 2004. ISSN 00078506. doi: 10.1016/S0007-8506(07)60721-5.

George Chryssolouris. *Manufacturing systems: theory and practice*. Springer Science & Business Media, 2013b.

George Chryssolouris, Konstantinos Efthymiou, Nikolaos Papakostas, Dimitris Mourtzis, and Aris Pagoropoulos. Flexibility and complexity: is it a trade-off? *International Journal of Production Research*, 51(23-24):6788–6802, 2013. ISSN 0020-7543. doi: 10.1080/00207543.2012.761362.

P Cilliers and D Spurrett. Complexity and post-modernism: Understanding complex systems. *South African Journal of Philosophy*, 18(2):258–274, 1999.

Paul Cilliers. *A framework for understanding complex systems*. 2004.

John Clarkson and Claudia Eckert. *Design process improvement: a review of current practice*. Springer Science & Business Media, 2010.

Simon Collinson and Melvin Jay. From Complexity to Simplicity: Unleash your organization's potential. pages 1–300, 2012. doi: 10.1057/9781137006226.

International Electrotechnical Commission et al. Iec 62264-1 enterprise-control system integration–part 1: Models and terminology. *IEC, Genf*, 2003.

Peter a Corning. Complexity Is Just a Word! *Technological Forecasting and Social Change*, 59(2):197–200, 1998. ISSN 00401625. doi: 10.1016/S0040-1625(97) 00130-3.

Thomas M. Cover and Joy A. Thomas. Elements of Information Theory 2nd Edition. page 776, 2006. ISSN 15579654. doi: 10.1007/978-94-010-9292-0.

J. G. Dai. On Positive Harris Recurrence of Multiclass Queueing Networks: A Unified Approach Via Fluid Limit Models. *The Annals of Applied Probability*, 5 (1):49–77, 1995. ISSN 1050-5164. doi: 10.1214/aoap/1177004828.

Wenbin Dai and Valeriy Vyatkin. A Component-Based Design Pattern for Improving Reusability of Automation Programs. *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*, pages 4328–4333, 2013.

Ahmed M. Deif and Hoda a. ElMaraghy. Modelling and analysis of dynamic capacity complexity in multi-stage production. *Production Planning & Control*, 20 (8):737–749, 2009. ISSN 0953-7287. doi: 10.1080/09537280903119072.

Abhijit V Deshmukh. *Complexity and chaos in manufacturing systems*. PhD thesis, 1993.

Abhijit V Deshmukh, Joseph J Talavage, and Moshe M Barash. Complexity in Manufacturing Systems Part 1 : Analysis of Static Complexity 1 Introduction. *IIE Transactions*, 30:645 – 655, 1998. doi: 10.1023/A:1007542328011.

Peter Dewhurst and Nicholas Abbatiello. Design for service. In *Design for X*, pages 298–317. Springer, 1996.

Rong Ding and Yan-Ming Sun. An entropy approach to structural and operational evaluation of manufacturing resource organization modes. *Industrial Engineering Journal/Gongye Gongcheng*, 15(3):69–74, 2012.

R. Donner, B. Scholz-Reiter, and U. Hinrichs. Nonlinear characterization of the performance of production and logistics networks. *Journal of Manufacturing Systems*, 27(2):84–99, 2008. ISSN 02786125. doi: 10.1016/j.jmsy.2008.10.001.

B. Efatmaneshnik, M., Nilchiani, R., & Heydari. From complicated to complex uncertainties in system of systems. In *Systems Conference (SysCon), 2012 IEEE International.*, pages 1–6, 2012.

Janet Efstathiou, Ani Calinescu, and Guy Blackburn. A web-based expert system to assess the complexity of manufacturing organizations. *Robotics and Computer-Integrated Manufacturing*, 18(3-4):305–311, 2002. ISSN 07365845. doi: 10.1016/S0736-5845(02)00022-4.

K Efthymiou. *On the assessment of manufacturing systems complexity*. PhD thesis, University of Patras, 2013.

K Efthymiou, N Papakostas, D Mourtzis, and G Chryssolouris. Fluid Dynamics Analogy to Manufacturing Systems. In *42nd CIRP Conference on Manufacturing Systems*, 2009.

K. Efthymiou, a. Pagoropoulos, N. Papakostas, D. Mourtzis, and G. Chryssolouris. Manufacturing systems complexity review: Challenges and outlook. *Procedia CIRP*, 3:644–649, 2012. ISSN 22128271. doi: 10.1016/j.procir.2012.07.110.

K. Efthymiou, D. Mourtzis, A. Pagoropoulos, N. Papakostas, and George Chryssolouris. Manufacturing systems complexity analysis methods review. *International Journal of Computer Integrated Manufacturing*, 2016a.

Konstantinos Efthymiou, Aris Pagoropoulos, Nikolaos Papakostas, Dimitris Mourtzis, and George Chryssolouris. Manufacturing systems complexity : An assessment of manufacturing performance indicators unpredictability. *CIRP Journal of Manufacturing Science and Technology*, 7:324–334, 2014.

Konstantinos Efthymiou, Dimitris Mourtzis, A Pagoropoulos, Nikolaos Papakostas, and George Chryssolouris. Manufacturing systems complexity analysis methods review. *International Journal of Computer Integrated Manufacturing*, 29(9):1025–1044, 2016b.

W.H. ElMaraghy and R.J. Urbanic. Assessment of Manufacturing Operational Complexity. *CIRP Annals - Manufacturing Technology*, 53:401–406, 2004. ISSN 00078506. doi: 10.1016/S0007-8506(07)60726-4.

H ElMaraghy, G Schuh, W ElMaraghy, and F Piller. Product variety management. *CIRP Annals-Manufacturing Technology*, 62(2):629–652, 2013.

H. Elmaraghy, T. Algeddawy, S. N. Samy, and V. Espinoza. A model for assessing the layout structural complexity of manufacturing systems. *Journal of Manufacturing Systems*, 33(1):51–64, 2014. ISSN 02786125. doi: 10.1016/j.jmsy.2013.05.012.

H. A. ElMaraghy. Flexible and reconfigurable manufacturing systems paradigmsitle. *International journal of flexible manufacturing systems*, 17(4):261–276, 2005.

H. a. ElMaraghy, S. N. Samy, and V. Espinoza. A classification Code for Assembly Systems. *3rd CIRP Conference on Assembly Technologies and Systems*, 2010.

H.a. ElMaraghy, O. Kuzgunkaya, and R.J. Urbanic. Manufacturing Systems Configuration Complexity. *CIRP Annals - Manufacturing Technology*, 54(1):445–450, 2005. ISSN 00078506. doi: 10.1016/S0007-8506(07)60141-3.

Hoda a. ElMaraghy. A Complexity Code for Manufacturing Systems. In *ASME 2006 International Manufacturing Science and Engineering Conference*, volume 2006, pages 625–634, 2006. ISBN 0-7918-4762-4. doi: 10.1115/MSEC2006-21075.

Waguih Elmaraghy, Hoda Elmaraghy, Tetsuo Tomiyama, and Laszlo Monostori. Complexity in engineering design and manufacturing. *CIRP Annals - Manufacturing Technology*, 61(2):793–814, jan 2012. ISSN 00078506. doi: 10.1016/j.cirp.2012.05.001.

W.H. ElMaraghy and R.J. Urbanic. Modelling of Manufacturing Systems Complexity. *CIRP Annals - Manufacturing Technology*, 52:363–366, 2003. ISSN 00078506. doi: 10.1016/S0007-8506(07)60602-7.

WH ElMaraghy and RJ Urbanic. Assessment of manufacturing operational complexity. *CIRP Annals-Manufacturing Technology*, 53(1):401–406, 2004.

SD Eppinger. The design structure matrix-dsm. Technical report, Technical report, http://www. dsmweb. org, 2005.

Péter Érdi. *Complexity explained*. Springer Science & Business Media, 2008. ISBN 9783540357773. doi: 10.1007/978-3-540-35778-0.

V Espinoza, H ElMaraghy, T AlGeddawy, and S N Samy. Assessing the structural complexity of manufacturing systems layout. In *Proceedings of the 4th CIRP Conference on Assembly Technologies and Systems*, volume 18, pages 65–70, 2012. ISBN 978-0-615-64022-8.

A Falck, R Örtengren, and M Rosenqvist. Relationship between complexity in manual assembly work, ergonomics and assembly quality. In *NES2012 Ergonomics for Sustainability and Growth*, 2012.

Ann Christine Falck and Mikael Rosenqvist. What are the obstacles and needs of proactive ergonomics measures at early product development stages? - An interview study in five Swedish companies. *International Journal of Industrial Ergonomics*, 42(5):406–415, 2012. ISSN 01698141. doi: 10.1016/j.ergon.2012.05.002.

Ann-Christine Falck, Roland Örtengren, and Mikael Rosenqvist. Assembly failures and action cost in relation to complexity level and assembly ergonomics in manual assembly (part 2). *International Journal of Industrial Ergonomics*, 44 (3):455–459, 2014. ISSN 01698141. doi: 10.1016/j.ergon.2014.02.001.

Amro M Farid and Duncan C McFarlane. A design structure matrix based method for reconfigurability measurement of distributed manufacturing systems. *International Journal of Intelligent Control and Systems*, 12(2):118–129, 2007.

T Fässberg, Å Fasth, F Hellman, A Davidsson, and J Stahre. Interaction between complexity , quality and cognitive automation. *Proceedings of the 4th CIRP Conference on Assembly Technologies and Systems*, 2:145–150, 2012.

J D Fast. Entropy: The significance of the concept of entropy and its applications in science and technology. page 313, 1962.

Å Fast-Berglund, T. Fässberg, F. Hellman, a. Davidsson, and J. Stahre. Relations between complexity, quality and cognitive automation in mixed-model assembly. *Journal of Manufacturing Systems*, 32(3):449–455, 2013. ISSN 02786125. doi: 10.1016/j.jmsy.2013.04.011.

Marshall Fisher, Anjani Jain, and John Paul MacDuffie. Strategies for Product Variety: Lessons from the Auto Industry. In *Redesigning the Firm*, pages 116–154. 1995.

G Frizelle and Y Suhov. The measurement of complexity in production and other commercial systems. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 464:2649–2668, 2008. ISSN 1364-5021. doi: 10.1098/rspa.2007.0275.

G. Frizelle and Y. M. Suhov. An entropic measurement of queueing behaviour in a class of manufacturing operations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 457(2011):1579–1601, 2001. ISSN 1364-5021. doi: 10.1098/rspa.2000.0731.

G. Frizelle and E. Woodcock. Measuring complexity as an aid to developing operational strategy. *International Journal of Operations & Production Management*, 15(5):26–39, 1995. ISSN 0144-3577. doi: 10.1108/01443579510083640.

Hideo Fujimoto, Alauddin Ahmed, Yasuhiro Iida, and Mineo Hanai. Assembly process design for managing manufacturing complexities because of product varieties. *International Journal of Flexible Manufacturing Systems*, 15(4):283–307, 2003a.

Hideo Fujimoto, Alauddin Ahmed, Yasuhiro Iidda, and Mineo Hanai. Assembly process design for managing manufacturing complexities because of product varieties. *International Journal of Flexible Manufacturing Systems*, 15, 2003b. ISSN 09206299. doi: 10.1023/B:FLEX.0000036032.41757.3d.

Anthony Joseph Gabriel. The effect of internal static manufacturing complexity on manufacturing performance. (May), 2007a.

Anthony Joseph Gabriel. *The effect of internal static manufacturing complexity on manufacturing performance*. PhD thesis, Clemson University, 2007b.

T J Gabriel. Manufacturing Complexity : the Effects of Common Attributes of Manufacturing System Design on Performance. *Academy of Information and Management Sciences Journal*, 16(1):2013, 2013.

Ibrahim H. Garbie and Ashraf Shikdar. Design for Manufacturing Systems Complexity: A Perspective Approach. *ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis, Volume 4*, pages 751–762, 2010. doi: 10.1115/ESDA2010-25033.

Murray Gell-Mann. *The quark and the jaguar: adventures in the simple and the complex*. 1997. ISBN 0-7167-2725-0 (paperback). doi: 10.1119/1.18607.

N. Giannelos, N. Papakostas, D. Mourtzis, and G. Chryssolouris. Dispatching policy for manufacturing jobs and time-delay plots. *International Journal of Computer Integrated Manufacturing*, 20:329–337, 2007. ISSN 0951-192X. doi: 10.1080/09511920600786604.

Matthias Götzfried. *Managing Complexity Induced by Product Variety in Manufacturing Companies. Complexity Evaluation and Integration in Decision-Making*. PhD thesis, University of St. Gallen, 2013.

G. Grote. A participatory approach to the complementary design of highly automated work systems. *Human factors in organizational design and management–IV*, 1994.

Tor Guimaraes, Nils Martensson, Johan Stahre, and Magid Igbaria. Empirically testing the impact of manufacturing system complexity on performance. *International Journal of Operations & Production Management*, 19:1254–1269, 1999. ISSN 0144-3577. doi: 10.1108/01443579910294228.

P. Gullander, a. Davidsson, K. Dencker, Å. Fasth, T. Fässberg, U. Harlin, and J. Stahre. Towards a Production Complexity Model that Supports Operation, Rebalancing and Man-hour Planning. *Proceedings of the 4th Swedish Production Symposium (SPS)*, pages 1–10, 2011.

Ping Han, Li Li Ma, Wan Jiang Wu, and Chuang Liu. Research on influence factors and levels of automobile manufacturing system complexity. In *2011 IEEE 18th International Conference on Industrial Engineering and Engineering Management, IE and EM 2011*, number PART 1, pages 186–190. IEEE, sep 2011. ISBN 9781612844473. doi: 10.1109/IEEM.2011.6035136.

CT Herbst and H Herzel. Visualization of system dynamics using phasegrams. *Journal of The Royal Society Interface*, 10(85):2013, 2013.

C M Hinckley. A global conformance quality model. A new strategic tool for minimizing defects caused by variation, error, and complexity. *Stanford University.; TH: Thesis (Ph.D.); PBD: Jan 1994*, 1994.

Katja Holtta, Eun Suk Suh, Olivier de Weck, et al. Tradeoff between modularity and performance for engineered systems and products. In *ICED 05: 15th International Conference on Engineering Design: Engineering Design and the Global Economy*, page 2820. Engineers Australia, 2005.

S. J. Hu, X. Zhu, H. Wang, and Y. Koren. Product variety and manufacturing complexity in assembly systems and supply chains. *CIRP Annals - Manufacturing Technology*, 57:45–48, 2008. ISSN 00078506. doi: 10.1016/j.cirp.2008.03.138.

L. Huaccho Huatuco, J. Efstathiou, S. Sivadasan, and A. Calinescu. The value of dynamic complexity in manufacturing systems. In *International Conference of the Production and Operations Management Society*, pages 180–188, 2001.

Luisa Huaccho Huatuco, Janet Efstathiou, Ani Calinescu, Suja Sivadasan, and Stella Kariuki. Comparing the impact of different rescheduling strategies on the entropic-related complexity of manufacturing systems. *International Journal of Production Research*, 47(15):4305–4325, 2009. ISSN 0020-7543. doi: 10.1080/00207540701871036.

Heng Huang. *Facility layout using layout modules*. PhD thesis, 2003.

Erich Hückel. Quantentheoretische beiträge zum problem der aromatischen und ungesättigten verbindungen. iii. *Zeitschrift für Physik*, 76(9-10):628–648, 1932.

J.M. Jauregui-Becker, W.W. Wits, and F.J.a.M. Houten. Reducing design complexity of multidisciplinary domain integrated products: a case study. *Manufacturing Systems and Technologies for the New Frontier*, pages 149–154, 2008.

K. Jenab and D. Liu. A graph-based model for manufacturing complexity. *International Journal of Production Research*, 48(11):3383–3392, 2010. ISSN 0020-7543. doi: 10.1080/00207540902950860.

GAO Jianbo. Complex systems and emergence: How theory meets reality. *Ł*, 43 (4):359–389, 2013.

Michael Tiegelkamp Karl-Heinz John. Building Blocks of IEC 61131-3. Technical report, Geneva, 2003.

I. Katzorke and a. Pikovsky. Chaos and complexity in a simple model of production dynamics. *Discrete Dynamics in Nature and Society*, 5:179–187, 2000. ISSN 1026-0226. doi: 10.1155/S1026022600000510.

Konstantin Kernschmidt and Birgit Vogel-Heuser. An interdisciplinary sysml based modeling approach for analyzing change influences in production plants to support the engineering. In *Automation Science and Engineering (CASE), 2013 IEEE International Conference on*, pages 1113–1118. IEEE, 2013.

Anil Khurana. Managing Complex Production Processes. *Sloan Management Review*, 40(2):85–97, 1999.

Yong-Suk Kim. *A system complexity approach for the integration of product development and production system design*. Doctoral dissertation, Massachusetts Institute of Technology, 1999.

Matti J. Kinnunen. *Complexity Measures for System Architecture Models*. PhD thesis, Massachusetts Institute of Technology, 2006.

Witold Kinsner. System complexity and its measures: How complex is complex. In *Advances in cognitive informatics and cognitive computing*, pages 265–295. Springer, 2010.

G J Klir. Complexity: Some General Observations. *Systems Research*, 2:131–140, 1985. ISSN 07317239. doi: 10.1002/sres.3850020205.

Dominik Kohr, Lukas Budde, and Thomas Friedli. Identifying complexity drivers in discrete manufacturing and process industry. *Procedia CIRP*, 63:52–57, 2017.

H. Komoto and T. Tomiyama. Multi-disciplinary system decomposition of complex mechatronics systems. *CIRP Annals - Manufacturing Technology*, 60:191–194, 2011. ISSN 00078506. doi: 10.1016/j.cirp.2011.03.102.

Hitoshi Komoto and Tetsuo Tomiyama. A framework for computer-aided conceptual design and its application to system architecting of mechatronics products. *Computer-Aided Design*, 44(10):931–946, 2012.

Sergey Konstantinov, Mussawar Ahmad, Karthik Ananthanarayan, and Robert Harrison. The cyber-physical e-machine manufacturing system: Virtual engineering for complete lifecycle support. 2017.

Matthias Kreimeyer. A Structural Measurement System for Engineering Design Processes. page 420, 2009.

Kuzgunkaya and ElMaraghy. Assessing the structural complexity of manufacturing systems configurations. *International Journal of Flexible Manufacturing Systems*, 18(2):145–171, 2006.

AJD Fred Lambert and Surendra M Gupta. *Disassembly modeling for assembly, maintenance, reuse and recycling*. CRC press, 2016.

Jeff Lankford. Measuring system and software architecture complexity. In *Aerospace Conference, 2003. Proceedings. 2003 IEEE*, volume 8, pages 8_3849–8_3857. IEEE, 2003.

Robert B Laughlin. *A different universe: Reinventing physics from the bottom down*. Basic Books (AZ), 2005.

SR Lawrence and AH Buss. Shifting production bottlenecks: Causes, cures, and conundrums. *Production and Operations Management*, 3(1):21–37, 1994.

Jay Lee, Hung An Kao, and Shanhu Yang. Service innovation and smart analytics for Industry 4.0 and big data environment. In *Procedia CIRP*, volume 16, pages 3–8, 2014. ISBN 2212-8271. doi: 10.1016/j.procir.2014.02.001.

S M Lee, R Harrison, and a a West. A component-based control system for agile manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 219(1):123–135, 2005. ISSN 09544054. doi: 10.1243/095440505X8000.

SM Lee, R Harrison, AA West, and MH Ong. A component-based approach to the design and implementation of assembly automation system. *Proceedings of the*

Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 221(5):763–773, 2007.

T Lee. *Complexity theory in axiomatic design*. PhD thesis, Massachusetts Institute of Technology, 2003.

A. Lempel and J. Ziv. On the Complexity of Finite Sequences. *IEEE Transactions on Information Theory*, 22(1):75–81, jan 1976. ISSN 0018-9448. doi: 10.1109/TIT.1976.1055501.

K Li and P A Wieringa. Understanding perceived complexity in human supervisory control. *Cognition, Technology & Work*, 2(2):75–88, 2000. ISSN 1435-5558. doi: 10.1007/s101110050029.

Udo Lindemann, Maik Maurer, and Thomas Braun. *Structural Complexity Management: An Approach for the Field of Product Design*. 2008. ISBN 3540878890.

Haijiang Liu, Kaixiang Xu, and Zhenhua Pan. Modeling and application of mixed model assembly system complexity introduced by auto-body personalization. *The International Journal of Advanced Manufacturing Technology*, 2015a. ISSN 0268-3768. doi: 10.1007/s00170-015-7637-6.

Haijiang Liu, Kaixiang Xu, and Zhenhua Pan. Modeling and application of mixed model assembly system complexity introduced by auto-body personalization. *The International Journal of Advanced Manufacturing Technology*, pages 1–12, 2015b.

Jiansheng Liu, Haining Tu, Hua Zhang, Fangchen Xia, and Daoyuan Yu. Research on Measurement Entropy-Based of Equipment Management Complexity and Its Application in Production Planning. pages 604–611. Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-88518-4_65.

Peng Liu and Zhizhong Li. Task complexity: A review and conceptualization framework. *International Journal of Industrial Ergonomics*, 42(6):553–568, nov 2012. ISSN 01698141. doi: 10.1016/j.ergon.2012.09.001.

Niels Lohse. *Towards an Ontology Framework for the Integrated Design of Modular Assembly*. PhD thesis, University of Nottingham, 2006.

Ladislav Lukáš and Miroslav Plevný. Using entropy for quantitative measurement of operational complexity of supplier-customer system: case studies. *Central European Journal of Operations Research*, 24(2):371–387, 2016. ISSN 16139178. doi: 10.1007/s10100-015-0386-7.

Mark W Maier. Architecting principles for systems-of-systems. In *INCOSE International Symposium*, volume 6, pages 565–573. Wiley Online Library, 1996.

RM Maksimović and S Petrović. Complexity of production structures. *Facta universitatis-series: Mechanical Engineering*, 7:119–136, 2009.

a Makui and M B Aryanezhad. A new method for measuring the static complexity in manufacturing. *Journal of the Operational Research Society*, 54(5):555–557, 2003. ISSN 0160-5682. doi: 10.1057/palgrave.jors.2601493.

Hamed F Manesh and Dirk Schaefer. A virtual factory approach for design and implementation of agile manufacturing systems. In *2010 ASEE Annual Conference & Exposition*. University of Bath, 2010.

Michael Marti. *Complexity Management: Optimizing Product Architecture od Industrial Products*. Springer Science & Business Media, 2007.

Mark V Martin. Design for Variety. *Product Variety Management*, 1996.

Pierre Massotte. Behavioural analysis of a complex system. *The International Journal of Advanced Manufacturing Technology*, 12:66–76, 1996. ISSN 0268-3768. doi: 10.1007/BF01178963.

S. Mattsson, P. Gullander, U. Harlin, G. Bäckstrand, A. Fasth, and A. Davidsson. Testing complexity index - A method for measuring perceived production complexity. *Procedia CIRP*, 3(1):394–399, 2012. ISSN 22128271. doi: 10.1016/j.procir.2012.07.068.

Sandra Mattsson. *What is perceived as complex in final assembly? - To define, measure and manage production Complexity*. Phd thesis, Chalmers University of Technology, 2013.

Sandra Mattsson, Per Gullander, and Anna Davidsson. Method for measuring production complexity. *28th International Manufacturing Conference*, 2011a.

Sandra Mattsson, Per Gullander, and Anna Davidsson. Method for measuring production complexity. In *28th International Manufacturing Conference*, 2011b.

Sandra Mattsson, Malin Karlsson, Åsa Fast-Berglund, and Ida Hansson. Managing production complexity by empowering workers: Six cases. *Procedia CIRP*, 17: 212–217, 2014. ISSN 22128271. doi: 10.1016/j.procir.2014.02.041.

Sandra Mattsson, Malin Tarrar, and Åsa Fast-Berglund. Perceived production complexity understanding more than parts of a system. *International Journal of Production Research*, pages 1–9, 2016a. ISSN 0020-7543. doi: 10.1080/00207543.2016.1154210.

Sandra Mattsson, Malin Tarrar, and Åsa Fast-Berglund. Perceived production complexity–understanding more than parts of a system. *International Journal of Production Research*, 54(20):6008–6016, 2016b.

Maik S Maurer. *Structural awareness in complex product design*. PhD thesis, Universität München, 2007.

Thomas J McCabe. A complexity measure. *IEEE Transactions on software Engineering*, (4):308–320, 1976a.

T.J. McCabe. A Complexity Measure. *IEEE Transactions on Software Engineering*, SE-2, 1976b. ISSN 0098-5589. doi: 10.1109/TSE.1976.233837.

Charles S McLeod. *Development of a toolkit for component-based automation systems*. PhD thesis, Loughborough University, 2013.

Marc H Meyer and Alvin P Lehnerd. *The power of product platforms*. Simon and Schuster, 1997.

Marl H Meyer and Kathleen Foley Curlnotey. The impact of knowledge and technology complexity on information systems development. *Expert Systems With Applications*, 8(1):111–134, 1995.

Eve Mitleton-Kelly. *Complex Systems and Evolutionary Perspectives on Organizations: The Application of Complexity Theory to Organisations*. Elsevier Science Ltd, Oxford, UK, 2003. ISBN 0-08-043957-8.

S Miyakawa. The hitachi assemblablity evaluation method (aem). In *Proceedings of 1st Int. Conf. on Product Design for Assembly*, 1986.

V Modrak, S Bednar, and D Marton. Design Optimization of the Assembly Process Structure Based on Complexity Criterion. *Multimedia and Ubiquitous Engineering*, 2013.

Vladimir Modrak and Slavomir Bednar. Entropy Based versus Combinatorial Product Configuration Complexity in Mass Customized Manufacturing. *Procedia CIRP*, 41:183–188, 2016. ISSN 22128271. doi: 10.1016/j.procir.2015.12.100.

Matt Motyka, Jonathan R.a. Maier, and Georges M . Fadel. Representing the Complexity of Engineering Systems: A Multidisciplinary Perceptual Approach. *Proceedings of the Sixth International Conference on Complex Systems*, pages 564–571, 2006.

Dimitris Mourtzis, Michalis Doukas, and Foivos Psarommatis. Design and operation of manufacturing networks for mass customization. *CIRP Annals-Manufacturing Technology*, 62:467–470, 2013.

Vladimir Nikiforov. The energy of graphs and matrices. *Journal of Mathematical Analysis and Applications*, 326(2):1472–1475, 2007.

Shimon Y Nof, Wilbert Wilhelm, and H Warnecke. *Industrial assembly*. Springer Science & Business Media, 2012.

Gerhard Pahl and Wolfgang Beitz. *Engineering design: a systematic approach*. Springer Science & Business Media, 2013.

N. Papakostas and D. Mourtzis. An Approach for Adaptability Modeling in Manufacturing -Analysis Using Chaotic Dynamics. *CIRP Annals - Manufacturing Technology*, 56(1):491–494, 2007. ISSN 00078506. doi: 10.1016/j.cirp.2007.05.117.

N. Papakostas, K. Efthymiou, D. Mourtzis, and G. Chryssolouris. Modelling the complexity of manufacturing systems using nonlinear dynamics approaches. *CIRP Annals - Manufacturing Technology*, 58(1):437–440, 2009. ISSN 00078506. doi: 10.1016/j.cirp.2009.03.032.

Nikolaos Papakostas and George Chryssolouris. Manufacturing Systems Complexity an Assessment of. In *7th International Conference on Digital Enterprise Technology (Athens, Greece)*, number September, pages 383–391, 2011.

Kijung Park and Gül E. Okudan Kremer. Assessment of static complexity in design and manufacturing of a product family and its impact on manufacturing performance. *International Journal of Production Economics*, 169:215–232, 2015. ISSN 09255273. doi: 10.1016/j.ijpe.2015.07.036.

L Peliti and A Vulpiani. Measures of complexity. *Measures of Complexity*, 1988.

Marco Perona and Giovanni Miragliotta. Complexity management and supply chain performance assessment. A field study and a conceptual framework. *International Journal of Production Economics*, 90:103–115, 2004. ISSN 09255273. doi: 10.1016/S0925-5273(02)00482-6.

K Peters, J Worbs, U Parlitz, and H-P Wiendahl. Manufacturing systems with restricted buffer sizes. *Nonlinear Dynamics of Production Systems*, pages 39–54, 2004.

Karl Petersen. Ergodic Theory. 1983.

Matthew Peterson, Gregory M Mocko, and Joshua D Summers. Evaluating and comparing functional and geometric complexity of products. In *ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 551–558. American Society of Mechanical Engineers, 2012.

Areejit Phukan, Manoj Kalava, and Vittal Prabhu. Complexity metrics for manufacturing control architectures based on software and information flow. *Computers & Industrial Engineering*, 49(1):1–20, aug 2005. ISSN 03608352. doi: 10.1016/j.cie.2005.01.005.

Corrado. Poli. *Design for manufacturing: a structured approach*, volume 1. Elsevier Science & Technology Books, 2001. ISBN 978-0-7506-7341-9. doi: 10.1016/S0261-3069(02)00108-5.

Yunqing Rao and Janet Efstathiou. Entropy-based measurement of manufacturing system complexity and its application in scheduling. *Jixie Gongcheng Xuebao(Chinese Journal of Mechanical Engineering)*, 42(7):8–13, 2006.

Craig Read. *Complexity Characteristics and Measurement within Engineering Systems*. Doctoral thesis, Loughborough University, 2008.

Eberhardt Rechtin. *Systems architecting: Creating and building complex systems*. Prentice Hall, 1991.

Miles Richardson, Gary Jones, Mark Torrance, and Thornm Thom Baguley. Identifying the task variables that predict object assembly difficulty. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 48(3):511–525, 2006. ISSN 00187208. doi: 10.1518/001872006778606868.

P Romano. How can fluid dynamics help supply chain management? *International Journal of Production Economics*, 118(2):463–472, 2009. ISSN 09255273. doi: 10.1016/j.ijpe.2008.12.011.

Y Rong-Yi and H Xiao-Jing. Phase space reconstruction of chaotic dynamical system based on wavelet decomposition. *Chinese Physics B*, 20(2), 2011.

S. N. Samy and H. ElMaraghy. A model for measuring products assembly complexity. *International Journal of Computer Integrated Manufacturing*, 23(11): 1015–1027, 2010a. ISSN 0951-192X. doi: 10.1080/0951192X.2010.511652.

S N Samy and H Elmaraghy. Parts and Assembly Equipment Complexity Dependency Matrix. *Enabling Manufacturing Competitiveness and Economic Sustainability*, pages 452–457, 2011.

S. N. Samy and H. ElMaraghy. A model for measuring complexity of automated and hybrid assembly systems. *International Journal of Advanced Manufacturing Technology*, 62:813–833, 2012. ISSN 02683768. doi: 10.1007/s00170-011-3844-y.

S N Samy, T Algeddawy, and H Elmaraghy. A granularity model for balancing the structural complexity of manufacturing systems equipment and layout. *Journal of Manufacturing Systems*, 36:7–19, 2015.

SN Samy and H ElMaraghy. A model for measuring products assembly complexity. *International Journal of Computer Integrated Manufacturing*, 23(11):1015–1027, 2010b.

M Sandri. Numerical calculation of Lyapunov exponents. *Mathematica Journal*, 6 (3):78–84, 1996.

Joseph Sarkis. An empirical analysis of productivity and complexity for flexible manufacturing systems. *International Journal of Production Economics*, 48:39–48, 1997. ISSN 09255273. doi: 10.1016/S0925-5273(96)00025-4.

H Schleich, J Schaffer, and LF F Scavarda. Managing complexity in automotive production. *19th International Conference on Production Research*, pages 1–6, 2003.

H Schleifenbaum, J Uam, G Schuh, and C Hinke. Turbulence in Production Systems Fluid Dynamics and its Contributions to Production Theory. II, 2009a.

H Schleifenbaum, JY Uam, G Schuh, and C Hinke. Turbulence in production systems–fluid dynamics and its contributions to production theory. In *Proceedings of the world congress on engineering and computer science*, volume 2, 2009b.

Sandro Luis Schlindwein and Ray Ison. Human knowing and perceived complexity: implications for systems practice. *Emergence: Complexity and Organization*, 6 (3):27–32, 2004.

J.P.M. Schmitz, D.a. van Beek, and J.E. Rooda. Chaos in discrete production systems? *Journal of Manufacturing Systems*, 21(3):236–246, 2002. ISSN 02786125. doi: 10.1016/S0278-6125(02)80164-9.

Florian Schoettl, Max Christian Paefgen, and Udo Lindemann. Approach for measuring change-induced complexity based on the production architecture. *Procedia CIRP*, 17:172–177, 2014. ISSN 22128271. doi: 10.1016/j.procir.2014.01.043.

B. Scholz-Reiter, M. Freitag, and a. Schmieder. Modelling and Control of Production Systems based on Nonlinear Dynamics Theory. *CIRP Annals - Man-*

ufacturing Technology, 51:375–378, 2002. ISSN 00078506. doi: 10.1016/S0007-8506(07)61540-6.

G Schuh and U Schwenk. Produktkomplexität managen–strategien. *Methoden, Tools, München, Wien*, 2001.

Günther Schuh, Thomas Gartzen, and Johannes Wagner. Complexity-oriented ramp-up of assembly systems. *CIRP Journal of Manufacturing Science and Technology*, 10:1–15, 2015. ISSN 17555817. doi: 10.1016/j.cirpj.2015.05.007.

Giovan B. Scuricini. Complexity in large technological systems. *Measures of Complexity*, 314:83–101, 1988. doi: 10.1007/3-540-50316-1.

Cristian Secchi, Marcello Bonfé, Cesare Fantuzzi, Roberto Borsari, and Davide Borghi. Object-oriented modeling of complex mechatronic components for the manufacturing industry. *IEEE/ASME Transactions on Mechatronics*, 12(6):696–702, 2007.

Cosma Rohilla Shalizi and James P. Crutchfield. Computational mechanics: Pattern and prediction, structure and simplicity. *Journal of Statistical Physics*, 104(3-4):817–879, 2001. ISSN 00224715. doi: 10.1023/A:1010388907793.

C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5:3, 2001. ISSN 15591662. doi: 10.1145/584091.584093.

Jingqiu Shao and Yingxu Wang. A new measure of software complexity based on cognitive weights. *Canadian Journal of Electrical and Computer Engineering*, 28(2):69–74, 2003.

Hirokazu Shibata, Brent Cheldelin, and Kosuke Ishii. Assembly Quality Methodology: A New Method for Evaluating Assembly Complexity in Globally Distributed Manufacturing. *Design Engineering, Volumes 1 and 2*, 2003:335–344, 2003. ISSN 15214613. doi: 10.1115/IMECE2003-42415.

PLM Siemens. Software, 2009. *Tecnomatix Jack, Siemens PLM Software, Plano, TX*, 2003.

Herbert a Simon. The Architecture of Complexity: Hierarchic Systems. *The Sciences of the Artificial*, 106(December 1962):467–482, 1996. ISSN 0003-049X. doi: 10.2307/985254.

Kaushik Sinha. *Structural complexity and its implications for design of cyber-physical systems*. PhD thesis, Massachusetts Institute of Technology, 2014.

Kaushik Sinha and Olivier L de Weck. Structural complexity metric for engineered complex systems and its application. In *Gain Competitive Advantage by Managing Complexity: Proceedings of the 14th International DSM Conference Kyoto, Japan*, pages 181–194, 2012.

Kaushik Sinha and Olivier L de Weck. Structural complexity quantification for engineered complex systems and implications on system architecture and design. *matrix*, 1:1, 2013.

Kaushik Sinha and Eun Suk Suh. Pareto-optimization of complex system architecture for structural complexity and modularity. *Research in Engineering Design*, 29(1):123–141, 2018.

Kaushik Sinha, Narek R Shougarian, and Olivier L de Weck. Complexity management for engineered systems using system value definition. In *Complex Systems Design & Management*, pages 155–170. Springer, 2017.

S. Sivadasan, J. Efstathiou, G. Frizelle, R. Shirazi, and A. Calinescu. An information-theoretic methodology for measuring the operational complexity of supplier-customer systems. *International Journal of Operations & Production Management*, 22(1):80–102, 2002. ISSN 0144-3577. doi: 10.1108/01443570210412088.

S. Sivadasan, J. Efstathiou, a. Calinescu, and L. Huaccho Huatuco. Advances on measuring the operational complexity of supplier-customer systems. *European Journal of Operational Research*, 171:208–226, 2006. ISSN 03772217. doi: 10.1016/j.ejor.2004.08.032.

S Sivadasan, J Smart, L Huaccho Huatuco, and a Calinescu. Operational complexity and suppliercustomer integration: case study insights and complexity rebound.

Journal of the Operational Research Society, 61(12):1709–1718, 2010. ISSN 0160-5682. doi: 10.1057/jors.2009.138.

J. Smart, A. Calinescu, and L. Huaccho Huatuco. Extending the information-theoretic measures of the dynamic complexity of manufacturing systems. *International Journal of Production Research*, 51(January 2013):1–18, 2013. ISSN 0020-7543. doi: 10.1080/00207543.2011.638677.

Ian Sommerville. Software engineering. international computer science series. *ed: Addison Wesley*, 2004.

RK Standish. On complexity and emergence. *arXiv preprint nlin/0101006*, 2001.

Stephen M Stigler. Gauss and the invention of least squares. *The Annals of Statistics*, pages 465–474, 1981.

Qiang Su, Lei Liu, and Daniel E Whitney. A systematic study of the prediction model for operator-induced assembly defects based on assembly complexity factors. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(1):107–120, 2010.

Nam P. Suh. Axiomatic Design Theory for Systems. *Research in Engineering Design*, 10:189–209, 1998. ISSN 0934-9839. doi: 10.1007/s001639870001.

Nam P Suh. Complexity in Engineering. *CIRP Annals Manufacturing Technology*, 54(2):46–63, 2005a. ISSN 00078506. doi: 10.1016/S0007-8506(07)60019-5.

NP Suh. *Complexity: theory and applications*. Oxford University Press on Demand, 2005b.

Antônio Márcio Tavares Thomé and Rui Sousa. Design-manufacturing integration and manufacturing complexity. *International Journal of Operations & Production Management*, 36(10):1090–1114, oct 2016. ISSN 0144-3577. doi: 10.1108/IJOPM-11-2014-0550.

Andrew Harold Tilstra, Carolyn Conner Seepersad, and Kristin L Wood. A high-definition design structure matrix (hddsm) for the quantitative assessment of product architecture. *Journal of Engineering Design*, 23(10-11):767–789, 2012.

T Tomiyama. Dealing with Complexity in Design: A Knowledge Point of View. *Design Methods for Practice*, pages 137–146, 2006.

T. Tomiyama, V. D'Amelio, J. Urbanic, and W. ElMaraghy. Complexity of multidisciplinary design. *CIRP Annals-Manufacturing Technology*, 2007.

Karl Ulrich. The role of product architecture in the manufacturing firm. *Research Policy*, 24(3):419–440, may 1995. ISSN 00487333. doi: 10.1016/0048-7333(94)00775-3.

RJ Jill Urbanic and Waguih H. WH ElMaraghy. Modeling of Manufacturing Process Complexity. *Advances in Design*, pages 425–436, 2006.

Birgit Vogel-Heuser, Daniel Schütz, Timo Frank, and Christoph Legat. Model-driven engineering of manufacturing automation software projects–a sysml-based approach. *Mechatronics*, 24(7):883–897, 2014.

Supachai Vongbunyong and Wei Hua Chen. General disassembly process. pages 9–24, 2015.

R. Vrabič and P. Butala. Computational mechanics approach to managing complexity in manufacturing systems. *CIRP Annals - Manufacturing Technology*, 60(1):503–506, 2011. ISSN 00078506. doi: 10.1016/j.cirp.2011.03.050.

R. Vrabič and P. Butala. Assessing operational complexity of manufacturing systems based on statistical complexity. *International Journal of Production Research*, 50(14):3673–3685, 2012. ISSN 0020-7543. doi: 10.1080/00207543.2011.575098.

MM Waldrop. *Complexity - the Emerging Science At the Edge of Order and Chaos - Waldrop,Mm*. Simon and Schuster, 1993. ISBN 0671872346. doi: 10.1016/0024-6301(96)85336-2.

H. Wang and S. J. Hu. Manufacturing complexity in assembly systems with hybrid configurations and its impact on throughput. *CIRP Annals - Manufacturing Technology*, 59:53–56, 2010. ISSN 00078506. doi: 10.1016/j.cirp.2010.03.007.

He Wang, Xiaowei Zhu, S Jack Hu, Zhongqin Lin, and Guanlong Chen. Product Family Design to Minimize Manufacturing Complexity in Mixed-Model Assembly Systems. In *ASME 2009 International Manufacturing Science and Engineering Conference*, pages 525–534. ASME, 2009. ISBN 9780791843628 (ISBN). doi: 10.1115/MSEC2009-84045.

He Wang, Xiaowei Zhu, Hui Wang, S. Jack Hu, Zhongqin Lin, and Guanlong Chen. Multi-objective optimization of product variety and manufacturing complexity in mixed-model assembly systems. *Journal of Manufacturing Systems*, 30:16–27, 2011. ISSN 02786125. doi: 10.1016/j.jmsy.2011.03.002.

He Wang, Hui Wang, and S. Jack Hu. Utilizing variant differentiation to mitigate manufacturing complexity in mixed-model assembly systems. *Journal of Manufacturing Systems*, 32(4):731–740, 2013. ISSN 02786125. doi: 10.1016/j.jmsy.2013.09.001.

Hui Wang. *Product variety induced complexity and its impact on mixed-model assembly systems and supply chains*. PhD thesis, 2010.

K Wang, H Wee, S Gao, and S Chung. Production and inventory control with chaotic demands. *Omega*, 33(2):97–106, 2005. ISSN 03050483. doi: 10.1016/j.omega.2004.04.001.

G Weiss. Scheduling and Control of Manufacturing Systems – a Fluid Approach. *PROCEEDINGS OF THE ANNUAL ALLERTON CONFERENCE ON COMMUNICATION CONTROL AND COMPUTING*, pages 557–586, 1999.

Elaine J. Weyuker. Evaluating software complexity measures. *IEEE transactions on Software Engineering*, 14(9):1357–1365, 1988.

H. Wiendahl and H. Scheffczyk. Simulation based analysis of complex production systems with methods of nonlinear dynamics. *CIRP Annals-Manufacturing Technology*, 48(1):357–360, 1999. ISSN 09240136. doi: 10.1016/S0924-0136(03)00177-8.

H. P. Wiendahl and P. Scholtissek. Management and Control of Complexity in Manufacturing. *CIRP Annals - Manufacturing Technology*, 43(2):533–540, 1994. ISSN 17260604. doi: 10.1016/S0007-8506(07)60499-5.

HP Wiendahl. Belast ungsorientierte Fertigungssteuerung. 1987.

Kirsty Williamson. *Research methods for students and professionals: Information management and systems*. 2000. ISBN 0949060895.

K. Windt, T. Philipp, and F. Böse. Complexity cube for the characterization of complex production systems. *International Journal of Computer Integrated Manufacturing*, 21(2):195–200, mar 2008. ISSN 0951-192X. doi: 10.1080/09511920701607725.

Y. Wu, G. Frizelle, and J. Efstathiou. A study on the cost of operational complexity in customer-supplier systems. *International Journal of Production Economics*, 106(1):217–229, 2007a. ISSN 09255273. doi: 10.1016/j.ijpe.2006.06.004.

Y Wu, G Frizelle, and J Efstathiou. A study on the cost of operational complexity in customer supplier systems. *International Journal of Production Economics*, 106:217–229, 2007b. doi: 10.1016/j.ijpe.2006.06.004.

FE Yates. Complexity and the limits to knowledge. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, 235(5):201–204, 1978.

Luiza Zeltzer, Veronique Limère, Hendrik Van Landeghem, El-houssaine Aghezzaf, and Johan Stahre. Measuring complexity in mixed-model assembly workstations. *International Journal of Production Research*, 51(15):4630–4643, 2013. ISSN 0020-7543. doi: 10.1080/00207543.2013.783246.

Luiza Zeltzer, El-Houssaine Aghezzaf, and Veronique Limère. Workload balancing and manufacturing complexity levelling in mixed-model assembly lines. *International Journal of Production Research*, 55(10):2829–2844, 2017.

T. Zhang and J. Efstathiou. The complexity of mass customization systems under different inventory strategies. *International Journal of Computer Integrated Manufacturing*, 19:423–433, 2006. ISSN 0951-192X. doi: 10.1080/09511920500399011.

Zhifeng Zhang. Modeling complexity of cellular manufacturing systems. *Applied Mathematical Modelling*, 35(9):4189–4195, 2011. ISSN 0307904X. doi: 10.1016/j.apm.2011.02.044.

Zhifeng Zhang. Manufacturing complexity and its measurement based on entropy models. *International Journal of Advanced Manufacturing Technology*, 62(9-12):867–873, 2012. ISSN 02683768. doi: 10.1007/s00170-011-3872-7.

Xiaowei Zhu. *Modeling product variety induced manufacturing complexity for assembly system design*. PhD thesis, 2009.

Xiaowei Zhu, S Jack Hu, Yoram Koren, Samuel P Marin, and Ningjian Huang. Sequence planning to minimize complexity in mixed-model assembly lines. In *Assembly and Manufacturing, 2007. ISAM'07. IEEE International Symposium on*, pages 251–258. IEEE, 2007a.

Xiaowei Zhu, S. Jack Hu, Yoram Koren, and Samuel P. Marin. Modeling of Manufacturing Complexity in Mixed-Model Assembly Lines. *Journal of Manufacturing Science and Engineering*, 130(5):051013, 2008. ISSN 10871357. doi: 10.1115/1.2953076.

Xiaowei Zhu Xiaowei Zhu, S.J. Hu, Yoram Koren Yoram Koren, S.P. Marin, and Ningjian Huang Ningjian Huang. Sequence Planning to Minimize Complexity in Mixed-Model Assembly Lines. *2007 IEEE International Symposium on Assembly and Manufacturing*, 2007b. doi: 10.1109/ISAM.2007.4288481.

Alois Zoitl and Valeriy Vyatkin. IEC 61499 architecture for distributed automation: The "glass half full" view. *IEEE Industrial Electronics Magazine*, 3(4):7–22, 2009. ISSN 19324529. doi: 10.1109/MIE.2009.934789.